2019

American Option Pricing: An Accelerated Lattice Model with Intelligent Lattice Search

Qianru Shang
Technological University Dublin, College of Business, qianru.shang@tudublin.ie

Brian Byrne
Technological University Dublin, College of Business, brian.byrne@tudublin.ie

Recommended Citation

Shang, Qianru, and Brian Byrne. "American option pricing: An accelerated lattice model with intelligent lattice search." The Journal of Derivatives 27.1 (2019): 92-108.

American Option Pricing: *An Accelerated Lattice Model with Intelligent Lattice Search*

QIANRU SHANG AND BRIAN BYRNE

QIANRU SHANG
is a PhD candidate in finance at the College of Business at Technological University Dublin
in Dublin, Ireland.
jennifer.shang@mydit.ie

BRIAN BYRNE
is a lecturer in finance and MSc Finance coordinator at the College of Business at Technological University Dublin
in Dublin, Ireland.
brian.byrne@dit.ie

ABSTRACT: *The authors introduce to the literature an intelligent lattice search algorithm to efficiently locate the optimal exercise boundary for American options. Lattice models can be accelerated by incorporating intelligent lattice search, truncation, and dynamic memory. We reduce computational runtime from over 18 minutes down to less than 3 seconds to estimate a 15,000-step binomial tree where the results obtained are consistent with a widely acclaimed literature. Delta and implied volatility can also be accelerated relative to standard models. Lattice estimation, in general, is considered to be slow and not practical for valuing large books of options or for promptly rebalancing a risk-neutral portfolio. Our technique transforms standard binomial trees and renders them to be at least on par with commonly used analytical formulae. More importantly, intelligent lattice search can be tweaked to reach varying levels of accuracy with different step size, while conventional analytical formulae are less flexible.*

TOPICS: ***Options, derivatives****

*All articles are now categorized by topics and subtopics. **View at PM-Research.com.**

American option pricing can present challenges with trade-offs embedded along an accuracy-efficiency spectrum. Approximation methods necessarily produce error. The margin of that error must be weighed against the uptick in estimation speed. Geske and Johnson (1984), Bunch and Johnson (1992), Huang et al. (1996), Carr (1998), and Ju (1998) developed analytic approximations that were convergent in the sense that when additional terms were included, their respective techniques became increasingly accurate while less efficient. Zhu (2006) developed an exact solution in the form of a Taylor's series expansion, which contains infinitely many terms. The model, however, is not practicable in terms of speed (Medvedev and Scaillet 2010).[1] Barone-Adesi and Whaley (1987, BAW) developed typically faster approximation techniques. Ju and Zhong (1999, Ju-Zhong), based on BAW, served to improve the longer maturity options accuracy with little sacrifice in terms of efficiency. Li (2010) extended further Ju-Zhong by introducing an improved smoothing condition for American options. BAW, Ju-Zhong, and Li (2010) approaches were all found to share the limitation that pricing is not convergent to the "true" price. Including additional terms does not invariably produce greater accuracy (Fabozzi et al. 2016). Instead of applying a quadratic approximation, Bjerksund and Stensland (1993, Bjerksund93) simplify the optimal exercise strategy by assuming a unique flat boundary. The improved approximation is presented in Bjerksund and Stensland (2002,

[1] We are grateful to Prof. Zhu Song-Ping for providing his opinion on the Zhu (2006) model in terms of speed.

Bjerksund02), where a second bound is also introduced. In practice, their lower-bound approach represents an accurate and very computer efficient approximation to the true American option value. In this article, we compare our accelerated binomial model with these commonly used analytical approximations. We show that our model can be comparable with these analytical approximations in speed for option pricing, delta estimation, and implied volatility estimation.

In practice, traders are likely to prioritize speed if approximation only entails a small compromise in accuracy. Nimble responses are viewed as important for preserving profit margins and for timely hedging. Not surprisingly, then, approximation techniques are relied upon heavily despite limitations. Estimating a book of options requires some insight on how the respective parameter values are likely to be incorporated into a given model. This might not be as simple as it seems. Uniform performance of analytical models cannot be guaranteed. This typically means that analytic models are regularly benchmarked against numerical techniques. Cox, Ross, and Rubinstein (1979, CRR) developed a binomial approach. It is widely accepted that greater accuracy can be introduced by making the CRR lattice mesh finer or by, more prosaically, incorporating a larger number of steps. Broadie and Detemple (1996) used a 15,000-step binomial model to obtain "true values." The binomial model is therefore acknowledged as a reliable workhorse and serves to benchmark other techniques. This lattice method, however, is distinctly viewed as having substantial computational workloads, runtimes, and gluttonous memory requirements. Not surprisingly, many variants and extensions of the original CRR binomial model were developed to accelerate estimation. Leisen and Reimer (1996, LR) and Tian (1993) extended the CRR binomial model by changing the function of the increment and decrement of stock price at each step. The adaptive mesh method, proposed by Figlewski and Gao (1999), builds a strip of finer lattice over a substrate tree and can be adapted to a wide variety of options. Staunton (2005) proposed several modifications to the LR model by adapting curtailed ranges and Richardson extrapolation. This modified LR model was found to be more efficient than other approximation methods considered. Joshi (2009) used standard deviation truncation, smoothing, and Richardson extrapolation, which led to better performance than the modified LR model proposed in Staunton (2005). The Chen and Joshi (2012, Chen–Joshi) incorporated tolerance truncation, Black–Scholes smoothing, and Richardson Extrapolation. This model constituted the premier model relative to 220 lattice permutations evaluated previously in Joshi (2009). In this article, we compare our accelerated binomial model with Chen–Joshi and show that our model performs better than this leading benchmark tree model.

This article outlines a lattice search algorithm to rapidly locate the early exercise node in each column of a binomial model. This ideally should dispense with the standard blanket test to repetitively compare the relative magnitudes of the exercise and holding values at each node. The methodology in this article follows the literature that has focused on discerning a *continuous* early exercise boundary so that the tree can be cleanly delineated between exercising and holding regions. Knowing in advance the vicinity of the optimal exercise boundary reduces greatly the quantum of computation. Kim and Byun (1994) specify the optimal exercise boundary for an American put option written on a non-dividend-paying stock. Curran (1995) subsequently extended the Kim and Byun (1994) approach to American put options with continuous dividend yields y and proposed the Diagonal Method, which can efficiently locate the early exercise boundary. A stipulation of this model, however, is that the risk-free interest rate r necessarily exceeds (is inferior to) the dividend yield y for puts (calls). In reality, this implies the model cannot be guaranteed to work in every instance. Basso et al. (2002, 2004) developed the insights of Kim–Byun–Curran to discern a binomial approximation to the optimal exercise boundary. Areal and Rodrigues (2013) use the early exercise boundary theory of Curran (1995) to accelerate the binomial model for pricing American options with discrete dividends. In this article, we open the Kim–Byun–Curran boundary theory to include the wider subset of parameter inputs where the American put (call) valuation is *not* constrained by $r \geq y$ ($r \leq y$). Furthermore, we propose an accelerated CRR model, incorporating the intelligent lattice search algorithm based on revamped optimal boundary theory as well as two acceleration technologies, for efficiently pricing American options with unrestricted continuous dividends.

The remaining article is organized as follows. In the next section, optimal exercise boundary theory is reviewed and extended to make practicable intelligent lattice search and we introduce the latter. We then describe two acceleration technologies: truncation and dynamic memory, which are standard in the literature. and integrate two acceleration technologies with the intelligent lattice search algorithm to accelerate the CRR pricing process. Following that section, we evaluate the relative performances of our accelerated CRR model, with varying benchmarks including a conventional CRR model implementation, other tree models from a recent literature, and several commonly used analytical formulae. We develop several metrics that capture the relative efficiencies in terms of option pricing, delta estimation, and implied volatility estimation. In the final section, we tease out conclusions. The Appendix includes the proof of the propositions and theorems developed in the initial section.

EXTENDING KIM–BYUN–CURRAN: THE OPTIMAL EXERCISE BOUNDARY ADAPTED FOR UNRESTRICTED CONTINUOUS DIVIDENDS

Consider an American put option with an initial stock price S, strike price X, time to maturity T, risk-free interest rate r, continuous dividend yield y, and volatility σ, priced by a n-step binomial tree. We define the number of time steps as i, the number of upward steps as j, and $S_{(i,j)}$ and $V_{(i,j)}$ as the stock price and option price respectively. The lattice structure is presented in Exhibit 18 in the Appendix complete with (i, j) mapping. Kim and Byun (1994) give the definition of the stopping region ***S***, the continuation region ***C***, and the optimal exercise state B(i): all nodes in a binomial tree are divided into two groups, which fall on two different regions. The *stopping region* ***S*** is a series of nodes whose option values are equal to their exercise values, which can be given by

$$\boldsymbol{S} \equiv \{(\mathrm{i},\mathrm{j})|\mathrm{V}_{(i,j)} = \mathrm{X} - \mathrm{S}_{(i,j)}\} \tag{1}$$

The nodes that belong to the stopping region are called *stopping nodes*. In addition, the *continuation region* ***C*** is a set of nodes where options are worth more if they are held instead of exercised, which can be defined as

$$\boldsymbol{C} \equiv \{(\mathrm{i},\mathrm{j})|\mathrm{V}_{(i,j)} > \mathrm{X} - \mathrm{S}_{(i,j)}\} \tag{2}$$

In this case, the option values are equal to their *holding values* below:

$$\mathrm{V}_{(i,j)} = [p\mathrm{V}_{(i+1,j+1)} + (1-p)\mathrm{V}_{(i+1,j)}]R^{-1} \tag{3}$$

where *p* represents the risk-neutral probability of an upward movement and *R* = *exp*(*rT*/*n*). The nodes that belong to the continuation region are called *continuation nodes*. ***I*** is defined as the set of time steps at which there is at least one stopping node, which can be given by

$$\boldsymbol{I} \equiv \{\mathrm{i}|(\mathrm{i},\mathrm{j}) \in \boldsymbol{S}, 0 \le j \le i \le n\} \tag{4}$$

Based on ***I***, $\boldsymbol{I}_0 = \{\mathrm{i}|(\mathrm{i},\mathrm{j}) \in \boldsymbol{S}, 0 \le j \le i \le n-1\}$ (from the penultimate column back) is proposed, which will be used later. In addition, the *optimal exercise state* B(i) represents the *biggest* j at the ith column when (i, j) ϵ ***S*** for i ϵ ***I***, which can be defined as

$$\mathrm{B(i)} \equiv max\{\mathrm{j}|(\mathrm{i},\mathrm{j}) \in \boldsymbol{S},\ \mathrm{i} \in \boldsymbol{I}\} \tag{5}$$

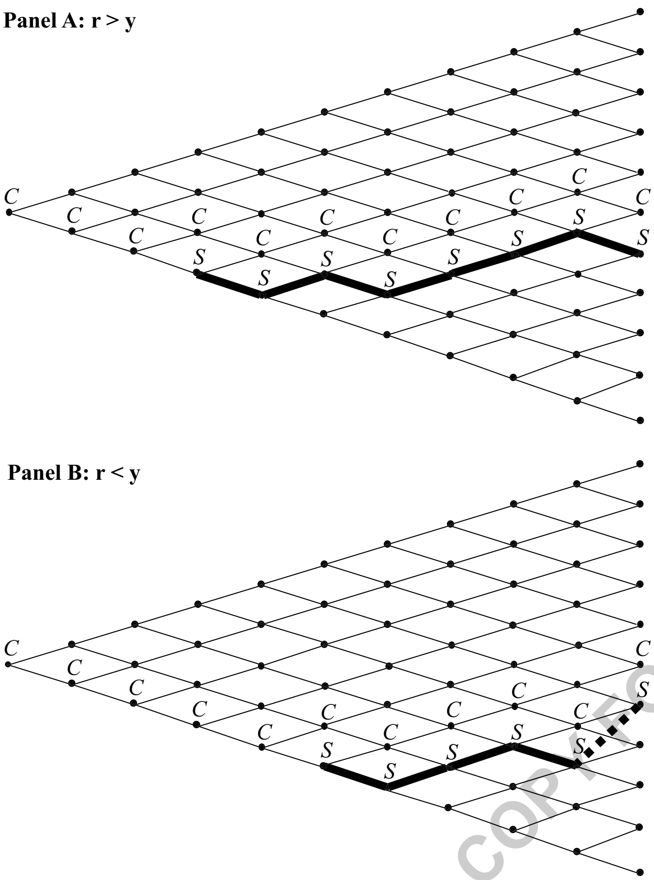
Therefore, (i, B(i)) is a series of *optimal exercise nodes* for i ϵ ***I***, which constitutes the *optimal exercise boundary*. Kim and Byun (1994) propose three propositions and two theorems relating to the continuity of the optimal exercise boundary. Curran (1995) extends the optimal exercise boundary theory for American put options with continuous dividend yields y where r ≥ y. He then applied this to American calls by invoking McDonald and Schroder (1998), who proposed put-call symmetry conditions for American options where

$$\mathrm{C(S,X,r,y,T,}\sigma) = \mathrm{P(X,S,y,r,T,}\sigma) \tag{6}$$

The optimal exercise boundary can also be applied for pricing American call options with y where r ≤ y. The Kim–Byun–Curran construction is augmented here by pricing American options without imposing any restrictions on y. The methodology developed in this article departs from Kim–Byun–Curran by locating/initializing in the penultimate column the seed node consistent with the optimal exercise boundary. The key intuition to the proposed approach relates to properties of the boundary. If r ≥ y the boundary is always continuous for put options, and this simplifies the demarcation of the stopping and continuation regions up to and including the final column. Otherwise, when r < y, a break of the early exercise boundary between the last column and

EXHIBIT 1

A Discontinuous Optimal Exercise Boundary When r < y



the penultimate column for an American put option can occur. In Exhibit 1, S represents the optimal exercise node (i, B(i)) for i ϵ ***I*** and C represents (i, B(i) + 1) for i ϵ ***I***, which is the *first* continuation node at each column from the bottom. The heavy solid lines represent the continuous optimal exercise boundary, and the heavy dashed lines represent the discontinuation. The upper binomial tree (Exhibit 1 [A]) follows that of Curran (1995, p.13), where S = 100, X = 100, T = 1, r = 0.05, y = 0, σ = 0.3 and n = 10. It is clear that the boundary is continuous where r > y. The lower binomial tree (Exhibit 1 [B]), however, has the same set of parameters as Curran's except y = 0.07, so that r < y (r = 0.05). In this instance, the stipulation that r ≥ y set out by Curran (1995) is violated. The impact of this violation is illustrated in the lower binomial tree. When r < y, the optimal exercise boundary is only continuous from the penultimate column back, while a discontinuous boundary between the penultimate and last column manifests itself.

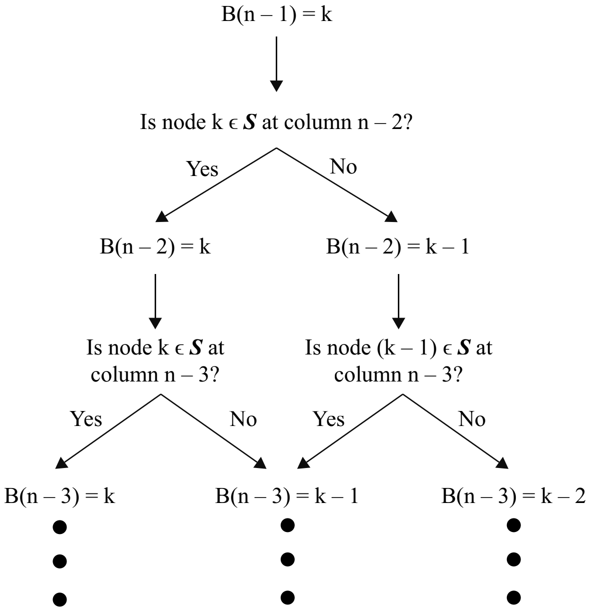
The main insight here is that the region of discontinuation is limited to merely the final and penultimate columns. If we exclude the final column, three propositions and two theorems of the optimal exercise boundary developed by Kim and Byun (1994) can be extended to American put options with unrestricted continuous dividends. These revamped propositions/theorems are developed in the Appendix. The restrictions imposed on dividends by Curran (1995) are also relaxed by seeding the continuous boundary from the penultimate column. The propositions and theorems in the Appendix can be used to identify the optimal exercise boundary for an American put option with unrestricted continuous dividend yield. We assert that *from the penultimate column back, the new optimal exercise state B(i − 1) is always equal to the old optimal exercise state B(i) minus a value of 1 or 0 as time to expiry increases.*[2] Discerning the adjustment behavior of the optimal exercise boundary permits an elaboration of an intelligent lattice search algorithm. Exhibit 2 shows the simple mechanism steering the intelligent lattice search, where this seed value at column n − 1 has been confirmed as an integer value of k. The optimal exercise state B(i − 1) is invariably equal to either B(i) or B(i) − 1. Therefore, we efficiently locate the boundary by verifying no more than one node at each column from the antepenultimate column back. Unlike Kim–Byun–Curran, the intelligent lattice search technique is anchored by reference to the penultimate column from where the recursion is initiated. This involves some further computation, as the exercise condition of additional nodes must be verified, (no more than n) at the penultimate column. The extra computation workload is comparatively trivial—especially for a large number of steps. In so doing, the restriction imposed on dividend yields can be relaxed and the spectrum of feasible parameter value inputs can be extended significantly.

TWO ACCELERATION TECHNOLOGIES: TRUNCATION AND DYNAMIC MEMORY

As noted by Curran (1995), there are subtrees within the binomial tree, where the nodes exercise no influence on the present value of the option. In the

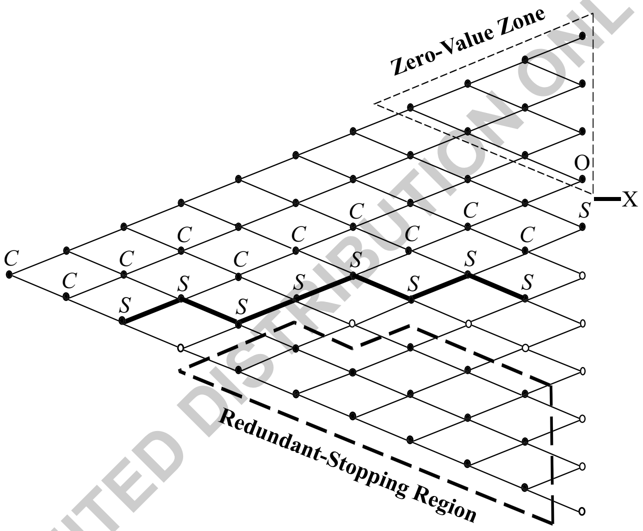
[2] Moving back through time—consistent with backward induction. Also, McDonald and Schroder (1998) put-call symmetry permits us to generalize to the case of the American call options.

EXHIBIT 2
An Intelligent Lattice Search Algorithm


B(n – 1) = k
Is node k ϵ *S* at column n – 2?
Yes
No
B(n – 2) = k
B(n – 2) = k – 1
Is node k ϵ *S* at column n – 3?
Is node (k – 1) ϵ *S* at column n – 3?
Yes
No
Yes
No
B(n – 3) = k
B(n – 3) = k – 1
B(n – 3) = k – 2

classic backward induction approach set out by CRR, all the nodes are given an equal weight in the estimation. This blanket method implies that large regions of the binomial tree are incorporated into the estimation but do not materially exert any impact in terms of the ultimate valuation. As noted by Curran (1995), this creates scope to apply the diagonal method where redundant nodes can be isolated and eliminated for the purposes of estimation.[3] Acceleration is obtained by locating and truncating a portion of redundant stopping nodes (The redundant-stopping region enclosed by heavy dashed lines) and all redundant zero-value nodes (The zero-value zone enclosed by light dashed lines), which are shown in Exhibit 3. Significantly, some redundant stopping nodes (hollow nodes) are not truncated because more-complex programming would be required for recognizing those nodes and this likely would produce slower computational speed. On balance, when the extra programming cost against speed is taken into account, it was considered sufficient to truncate the redundant-stopping region and the value of the redundant hollow nodes to make pricing more efficient.

EXHIBIT 3
Truncating Zero-Value Zone and Redundant-Stopping Region



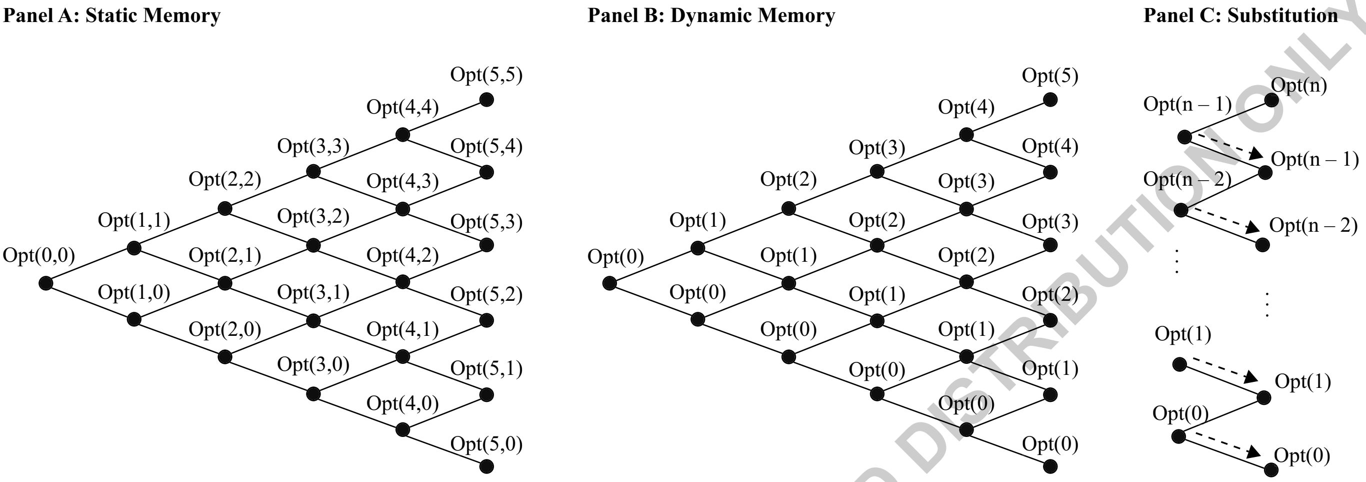
Making use of dynamic memory can produce an important reduction in computational cost. In Exhibit 4, we try to reveal how computer memory can be used more efficiently. A conventional two-dimensional static n-step binomial model requires (n + 1)(n + 2)/2 nodes to be memorized (Exhibit 4 (A)). Broadie and Detemple (1996) and Haug (2007, p. 288–289) propose using a one-dimensional dynamic binomial tree (Exhibit 4 [B]). This approach takes the option values at the last column and stores them in a dynamic vector Opt(j) for j = 0, 1, …, n. After moving one step back, the values in the re-dimensioned Opt(j) for j = 0, 1, …, n − 1 will be replaced by the option values of the corresponding nodes at the penultimate column (Exhibit 4 [C]). Similarly, the values of Opt(j) for j = 0, 1, …, k − 1 at kth column will always be substituted by the option values at (k − 1)th column for 1 ≤ k ≤ n. Therefore, a dynamic binomial tree only requires n + 1 contemporaneous storage spaces.

THE EFFICIENT PRICING PROCESS OF AN ACCELERATED CRR MODEL: APPLYING INTELLIGENT LATTICE SEARCH ALGORITHM, TRUNCATION, AND DYNAMIC MEMORY

In this section, we demonstrate how an accelerated CRR model, incorporating an intelligent lattice

[3] Consistent with Curran (1995), the first passage probabilities are not applied here. No increase in computational efficiency from that technique has been obtained.

Exhibit 4
A One-Dimensional Dynamic CRR Tree


Panel A: Static Memory **Panel B: Dynamic Memory** **Panel C: Substitution**

search algorithm, dynamic memory, and truncation, can efficiently price an American put option. We explain how the intelligent lattice search algorithm can be used to efficiently locate the optimal exercise boundary. We employ a one-dimensional dynamic binomial tree with truncation for an American put option. The rationale for presenting the sequence of steps involved in Exhibit 5 relates to teasing out a viable framework appropriate for coding. The CRR binomial tree in Exhibit 5 has the same set of parameters as Exhibit 1 (B), which also follows Curran (1995), where S = 100, X = 100, T = 1, r = 0.05, σ = 0.3 and n = 10. We differ by setting the dividend yield, y = 0.07. Since r < y, the stipulation that r ≥ y advanced by Curran (1995) is deliberately violated. The nodes falling along each row share the same underlying asset price and by extension exercise value. These are given in the final two columns. Array mappings in Exhibit 5 are set out consistent with the one-dimensional dynamic binomial tree depicted in Exhibit 4 (B). Each node outside the truncated regions can be identified by its series number. For illustrative purposes, nodes with series numbers enclosed by no-fill circles (e.g., ⓪) represent the stopping value. Nodes with series numbers enclosed by a black-fill circle (e.g., ⓿) represent the optimal exercise nodes (first stopping nodes), B(i), at ith column. Continuation nodes are in contrast denoted by square brackets (e.g., [0]). The nodes with series numbers followed by question marks are those that are minimally investigated by checking the exercise condition (e.g. ⓿? and [0]?). The optimal boundary check is efficiently reduced to determine the status of these nodes—no more than one single node at each column from the antepenultimate column. The pathway of these checks is shown by the presence of question marks. In this regard, clear efficiency gains are discernible vis-à-vis more common systems of blanket checking. The nodes represented by X without option values and series numbers in Exhibit 5 are redundant and can be truncated to also increase efficiency.

To elaborate the sequence of steps involved in the optimization process, we begin with the last column. The expression in (7) below is used to ascertain the first non-zero/stopping node at the maturity:

$$j = [(\ln(X/S)/\ln(u) + n)/2] \qquad (7)$$

where [.] locally means the largest integer lower than its argument and $u = \exp(\sigma\sqrt{T/n})$. Logically, for a put the nodes beneath the optimal exercise node at the maturity belong to the stopping region. Their exercise values are calculated and assigned to the appropriate nodes.[4] In Exhibit 5, j is initially calculated to be 4 using the expression in (7), which implies that the optimal exercise node at the maturity is node ❹. Then we assign respectively the exercise values:

[4] All assignments of the option values to nodes should strictly follow the order, from the bottom to the top, consistent with the dynamic memory approach.

Exhibit 5

An Efficient American Put Option Pricing Process

												Exercise Value	Underlying Price
											X	0.0000	258.2307
										X		0.0000	234.8590
									X		X	0.0000	213.6025
								X		X		0.0000	194.2699
							X		X		X	0.0000	176.6871
						X		X		X		0.0000	160.6956
					0.3898[4]		X		X		X	0.0000	146.1515
				1.6433[3]		0.7334[4]		X		X		0.0000	132.9237
			4.0259[2]		2.7516[3]		1.3797[4]		X		X	0.0000	120.8931
		7.5286[1]		6.1410[2]		4.5371[3]		2.5956[4]		X		0.0000	109.9514
	11.9473[0]		10.6532[1]		9.1540[2]		7.3327[3]		4.8830[4]		X	0.0000	100.0000
		15.9121[0]		14.6874[1]		13.2653[2]		11.5319[3]		9.1864[4]?		9.0507	90.9493
			20.6467[0]		19.6498[1]		18.5625[2]		17.4373[3]		17.2823❹	17.2823	82.7177
				26.0364[0]		25.4009[1]		24.8667[2]?		24.7949[3]?		24.7689	75.2311
					31.8493[0]?		31.6017[1]?		31.5778❷?		31.5778③	31.5778	68.4222
						37.7705⓿?		37.7705❶		37.7705❷?		37.7705	62.2295
							43.4028⓿		43.4028①		43.4028②	43.4028	56.5972
								X		48.5252①		48.5252	51.4748
									X		53.1841①	53.1841	46.8159
										X		57.4213	42.5787
											61.2749⓪	61.2749	38.7251
Column	0	1	2	3	4	5	6	7	8	9	10		

61.2749, 53.1841, 43.4028, 31.5778, and 17.2823, from node ⓪ to node ❹ at column 10.

Once these values are established, they then are used to enable backward induction leading to the penultimate column. The exercise condition is investigated somewhat more painstakingly from the first node beneath the zero-value zone by calculating and comparing the exercise values and holding values of the nodes until the optimal exercise node is confirmed. Then the respective exercise values are calculated again and assigned to the optimal exercise node and the node immediately below it, while consecutive holding values are calculated again and assigned to the nodes lying between the stopping zone and the zero-value zone. A more exhaustive search routine is required for the penultimate array because *the boundary is not guaranteed to be continuous going from the final column to the preceding column*. In Exhibit 5, moving to column 9 we check the exercise condition from node [4] by comparing the exercise values relative to the holding values of the nodes until node ❷, which is ultimately confirmed as the optimal exercise node. The respective exercise values, 48.5252 and 37.7705, are assigned to node ① and to node ❷. The respective holding values, 24.7949 and 9.1864, are assigned to node [3] and node [4]. By locating and verifying the optimal exercise value, the seed value of the continuous portion pertaining to the optimal exercise boundary, B(n − 1) is also identified. Node ❷ at column 9 provides the root value that initiates the continuous optimal boundary.

Then we move to column 8. Node ❷ should be initially inspected (the uncertain node) since it has the same series number as the optimal exercise node ❷ in column 9. We check its exercise condition and find that it is determined as a stopping node. This indicates that it is the optimal exercise node at this column. Thereafter, exercise values are assigned to it and the nodes immediately below it, and respective holding values are assigned to the nodes lying between the stopping zone and the zero-value zone at this column. Accordingly, we assign respective exercise values, 43.4028 and 31.5778, to node ① and node ❷ in column 8. The respective holding values, 17.4373 and 4.8830, are by default assigned to node [3] and node [4]. When we move to column 7, we check node [2], which has the same series number as the optimal exercise node ❷ in column 8 and find that node [2] is a continuation node, which means the node immediately below it, node ❶, is the optimal exercise node in column 7. Accordingly, we assign the exercise

Exhibit 6

The Closed-Loop Optimal Exercise Boundary Search Routine

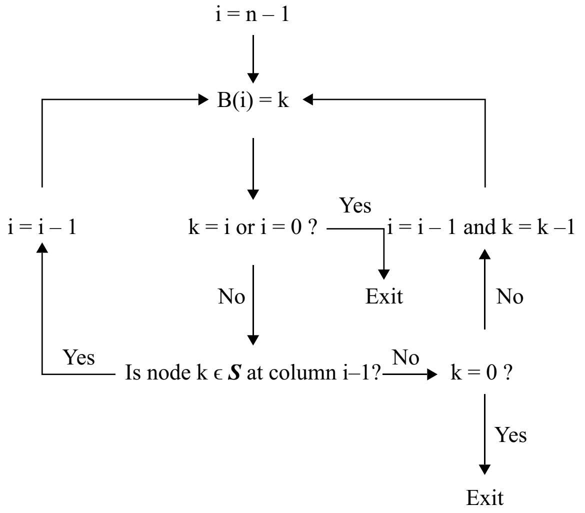


Exhibit 7

Comparing the Computational Workload Associated with the Accelerated CRR Tree and a Standard CRR Tree

	Number of Nodes	
Type of Nodes	**Accelerated CRR**	**Standard CRR**
Truncated Nodes	24	0
Stopping Nodes	9	11
Continuation Nodes	25	0
Uncertain Nodes	8	55
Total Nodes	66	66

Notes: Truncated nodes are ignored for backward recursion. Stopping nodes require merely to estimate the exercise value. Continuation nodes require merely to estimate the holding value. Uncertain nodes require both estimation of the exercise value and the holding value plus a logical operator to make a mutual comparison.

value 37.7705 to node ❶ and assign respective holding values, 24.8667, 11.5319, and 2.5956, to node [2], [3], [4] in column 7. This option valuing process is iterated until we move to column 4 by virtue that the value [0] associated with the last node in column 4, is also verified as a continuation node, which triggers the exit mechanism of the algorithm. The optimal exercise boundary searching stops. All nodes in the remaining columns (column 3, 2, 1, 0) are continuation nodes. From column 4 back, we assign automatically holding values to each node in each column.

We also present a closed-loop optimal exercise boundary search routine in Exhibit 6. When the previous exercise state B(i) = k, the new optimal exercise state either remains unchanged or minus 1 (B(i − 1) = k or k − 1). k should not breach 0 or exceed the step size associated with any given column in the tree. Otherwise, the exit feature is primed to trigger. In Exhibit 7, a direct comparison using the accelerated CRR tree vis-à-vis a standard CRR tree is made. The reported exhibit is based on the sample tree outlined in Exhibit 5. In a conventional CRR tree, we are normally obliged to estimate and assess the exercise value relative to holding value for 55 nodes. 11 terminal nodes are by default exercise values. The accelerated CRR tree only requires 8 direct comparisons to be made of the exercise value relative to the holding value. 9 nodes and 25 nodes are automatically assigned exercise values and holding values independently given that the early exercise boundary can be used to efficiently demarcate. Otherwise, 24 redundant nodes are truncated, which incorporates hardly any processing costs.

NUMERICAL RESULTS

Numerical results can be divided into three sections: In the first section, we show how option pricing efficiency can be improved by applying dynamic memory, truncation, and intelligent lattice search sequentially to a standard CRR tree. The most accelerated CRR model combines intelligent lattice search, dynamic memory, and truncation together. In the second section, we compare the efficiency of our most accelerated CRR model to a standard CRR model, to a leading benchmark tree Chen and Joshi (2012), and to four popular analytical formulae. These comparisons are made relative to both option pricing and delta estimation. In the final section, we compare the accelerated CRR model to Chen–Joshi, and to four analytical formulae for implied volatility estimation. All reported results are obtained using Excel VBA. A DELL Latitude E5470 with Intel's Core i3 processors ran these algorithms and models.

In the first section, we gauge successively how improvements in estimation efficiency can be introduced by using dynamic memory, intelligent lattice search, and truncation, where the initial baseline tree is

EXHIBIT 8

Acceleration Effect Comparison Among Dynamic Memory, Truncation, and Intelligent Lattice Search

Option				CRR_Dyn_	CRR_Dyn_	Broadie &
Param	**n**	**CRR**	**CRR_Dyn**	**Bound**	**Bound_Trun**	**Detemple**
	1000	12.1456	12.1456	12.1456	12.1456	
Amer		00:05.07	00:02.70	00:00.09	00:00.02	
Call	**3000**	12.1455	12.1455	12.1455	12.1455	
S = 80		00:43.09	00:22.13	00:00.67	00:00.33	
X = 100	**5000**	12.1452	12.1452	12.1452	12.1452	
T = 3		01:58.79	01:00.91	00:01.82	00:00.69	
r = 0.07	**10000**	12.1455	12.1455	12.1455	12.1455	
σ = 0.30		07:58.54	04:02.83	00:06.34	00:01.18	
y = 0.03	**15000**	12.1453	12.1453	12.1453	12.1453	12.145
		18:06.41	09:23.60	00:13.88	00:02.30	
	1000	9.2486	9.2486	9.2486	9.2486	
Amer		00:05.07	00:02.73	00:00.12	00:00.02	
Put	**3000**	9.2499	9.2499	9.2499	9.2499	
S = 100		00:42.70	00:22.08	00:00.84	00:00.39	
X = 100	**5000**	9.2502	9.2502	9.2502	9.2502	
T = 0.5		01:59.00	01:01.25	00:01.87	00:00.73	
r = 0.03	**10000**	9.2504	9.2504	9.2504	9.2504	
σ = 0.30		07:57.43	04:07.90	00:07.25	00:02.31	
y = 0.07	**15000**	9.2505	9.2505	9.2505	9.2505	9.251
		18:51.19	09:34.87	00:16.67	00:05.66	

Notes: CRR is a standard two-dimensional static CRR tree. CRR_Dynamic is a one-dimensional dynamic CRR tree. CRR_Dyn_Boundary is a dynamic CRR tree incorporating intelligent lattice search algorithm. CRR_Dyn_Bound_Truncation is a dynamic CRR tree incorporating intelligent lattice search and truncation, the most accelerated CRR model.

a standard CRR tree.[5] Benchmark values for American call and put option samples are obtained from Broadie and Detemple (1996). In addition, in order to test the intelligent lattice search algorithm, two sets of parameters with r > y (r < y) for call (put) are expressly selected, which violate the restrictions imposed by Curran (1995) on continuous dividend yields. In Exhibit 8, CRR baseline represents a standard two-dimensional static CRR tree. CRR_Dyn are accelerated purely by employing a one-dimensional dynamic tree. CRR_Dyn_Bound augments the dynamic binomial tree by using the intelligent lattice search algorithm. CRR_Dyn_Bound_Trun represents the most accelerated binomial model, which comprehensively applies intelligent lattice search, dynamic memory, and truncation. The computational times are presented using a mm:ss.00 format, located under the corresponding option values. We found that all four binomial models with the same set of parameters have resolutely identical results, which in turn are also consistent with the benchmark values. The acceleration effects can be gauged by noting how estimation time is reduced—moving from the baseline. Replacing the two-dimensional static tree by a one-dimensional dynamic tree saved almost half of the computational time. This pales in comparison to the acceleration effect of applying intelligent lattice search algorithm, which produces improvements in speed by at least one order of magnitude. Then the application of truncation technology further speeds up the computation several times. With the number of steps increasing, the effect of accelerations becomes more obvious. The baseline binomial model took more than 18 minutes at a 15,000-step size to complete. The most accelerated CRR tree (CRR_Dyn_Bound_Trun) took only 2.30 seconds for the call and 5.66 seconds for the put. The improvement in estimation time is noteworthy, and accuracy has not been compromised relative to the standard CRR tree.

[5] This approach to pricing is by far the slowest. It is also typically the most common method introduced in textbooks.

Before moving to the second section, there is some initial preparatory work relating to generating a large number of sample options parameters. We use these to determine the level of error relative to a benchmark ("true value") and the time required for estimation is also recorded. We additionally sketch out a number of delta estimation methods. Following Broadie and Detemple (1996), we design a uniform distribution of parameters inputs for S, T, r, y, σ, and PutCall to generate 2,500 American options. The spot price S was given to be uniformly distributed between 70 and 130. The exercise price X was fixed as a constant at a value of 100. Time to maturity T, with probability of 0.75, was uniform between 0.1 and 1 years. A probability of 0.25 was attributed to maturity being randomly between 1 and 5 years. The riskless rate r was uniformly distributed between 0 and 0.1 with a probability of 0.8 and 0 generated, with the residual probability of 0.2. The dividend rate y was uniform between 0 and 0.1. Volatility σ was distributed uniformly between 0.1 and 0.6. There was also a random 0.5 probability of the option being a call or put. Consistent with Broadie and Detemple (1996), the main error measure, root-mean-square relative error (RMSRE), is defined as

$$\text{RMSRE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m} e_i^2} \tag{8}$$

where m is the number of options and $e_i = \frac{\widehat{C_i} - C_i}{C_i}$ where C_i and $\widehat{C_i}$ is the true and estimated value of the option, respectively. The true value C_i is generated using a 15,000-step CRR model. 170 of the 2,500 American options with extremely-small true value ($C_i < 0.50$) are excluded. The residual number of American options involved in the valuation is 2,330 (m = 2,330). The time consumption measure (Time) represents the average execution time (seconds) for pricing per American option, which can be calculated as

$$\text{Time} = \frac{\textit{Total Execution Time}}{m} \tag{9}$$

where m is the number of options. For delta estimation, the delta of the tree model, including a standard CRR, our accelerated CRR, and Chen–Joshi, are estimated using

$$\Delta_{CRR} = \frac{V_{(1,1)} - V_{(1,0)}}{S_{(1,0)} - S_{(1,1)}} \tag{10}$$

where the numerator is the difference between the option value of the upper node and lower node at the end of the first period, and the denominator is the difference between the stock price of these two nodes. For analytical formulae, however, the delta is calculated as

$$\Delta_{anal} = \frac{C(S+\varepsilon) - C(S-\varepsilon)}{2\varepsilon} \tag{11}$$

where ε is a perturbation introduced for the spot price S, and the numerator is the difference between the option value estimated with the spot price $S + \varepsilon$ and $S - \varepsilon$ using analytical formulae.

In the second section, we first compare the 2,330 options sample for pricing and delta estimation, with a view to teasing out the relative efficiency of a standard CRR model (CRR) vis-à-vis our accelerated CRR model (Accel CRR). The latter introduces intelligent lattice search, dynamic memory, and truncation. For both pricing options and estimating delta, shown in Exhibit 9, as the number of steps increase, the estimation error (RMSRE) generated by CRR and our Accel CRR decreases, while the execution time (Time) increases. We found that Accel CRR always generates identical RMSRE as CRR at different step size but with much less Time, which implies that the pricing process is effectively accelerated without disturbing the accuracy. The "Multiple of speed" shows how many times Accel CRR is faster than CRR attaining the same estimation accuracy. From 50 to 1,000 steps, Accel CRR model can be from 7 to 220 (182) times faster than CRR in option pricing (estimating delta).

To provide a yardstick relative to a more recent literature, we replicate Chen–Joshi and run it with a tolerance level of 1E-05 to estimate 2,330 generated sample options. The comparison between our Accel CRR and Chen–Joshi is demonstrated in Exhibit 10. To make a direct comparison, the number of steps of the two models are selected to achieve a similar level of accuracy (RMSRE) so that the efficiency can be easily juxtaposed according to execution time (Time). For each column, Accel CRR and Chen–Joshi generate a similar RMSRE but expend different amounts of Time. The "Multiple of speed" indicates how many times Accel CRR is faster than Chen–Joshi. For option pricing, Accel CRR is roughly 1.5 to 2 times faster than Chen–Joshi consistent with a similar level of accuracy.

EXHIBIT 9

Comparing Accel CRR and CRR in Option Pricing and Delta Estimation

		Steps	50	100	200	300	400	500	600	700	800	900	1000
Option Pricing	Accel CRR	RMSRE	7.65E-03	3.61E-03	1.79E-03	1.08E-03	8.05E-04	6.85E-04	5.71E-04	4.64E-04	4.10E-04	3.67E-04	3.85E-04
		Time (T_a)	1.77E-03	2.91E-03	3.42E-03	6.72E-03	8.06E-03	9.18E-03	1.01E-02	1.25E-02	1.45E-02	1.75E-02	2.20E-02
	CRR	RMSRE	7.65E-03	3.61E-03	1.79E-03	1.08E-03	8.05E-04	6.85E-04	5.71E-04	4.64E-04	4.10E-04	3.67E-04	3.85E-04
		Time (T_0)	1.27E-02	4.84E-02	1.72E-01	3.79E-01	6.60E-01	1.0433	1.4884	2.0246	2.6490	3.8083	4.8481
	Multiple of Speed (T_0/T_a)		7.19	16.61	50.34	56.37	81.97	113.59	147.32	162.16	182.77	217.75	220.37
Delta Estimation	Accel CRR	RMSRE	7.34E-03	3.55E-03	1.76E-03	1.09E-03	8.32E-04	7.14E-04	5.92E-04	4.95E-04	4.40E-04	3.95E-04	4.01E-04
		Time (T_a)	1.62E-03	2.31E-03	3.71E-03	5.39E-03	8.04E-03	9.26E-03	1.13E-02	1.36E-02	1.62E-02	1.88E-02	2.18E-02
	CRR	RMSRE	7.34E-03	3.55E-03	1.76E-03	1.09E-03	8.32E-04	7.14E-04	5.92E-04	4.95E-04	4.40E-04	3.95E-04	4.01E-04
		Time (T_0)	1.13E-02	4.70E-02	1.69E-01	3.81E-01	6.65E-01	1.0371	1.4815	1.9546	2.5505	3.2325	3.9707
	Multiple of Speed (T_0/T_a)		7.01	20.37	45.57	70.67	82.66	111.98	130.56	143.26	157.42	172.12	182.08

EXHIBIT 10

Comparing Accel CRR and Chen–Joshi in Option Pricing and Delta Estimation

		Steps	30	120	200	300	400	500	600	700	900	1100
Option Pricing	Accel CRR	RMSRE	1.08E-02	2.88E-03	1.79E-03	1.08E-03	8.05E-04	6.85E-04	5.71E-04	4.64E-04	3.67E-04	3.13E-04
		Time (T_a)	1.46E-03	3.22E-03	3.42E-03	6.72E-03	8.06E-03	9.18E-03	1.01E-02	1.25E-02	1.75E-02	2.42E-02
		Steps	10	20	30	40	50	60	70	80	90	100
	Chen–Joshi	RMSRE	1.14E-02	2.87E-03	1.39E-03	8.60E-04	6.26E-04	5.72E-04	4.58E-04	3.38E-04	3.06E-04	3.04E-04
		Time (T_0)	2.81E-03	4.76E-03	7.63E-03	1.12E-02	1.53E-02	2.06E-02	2.60E-02	3.27E-02	3.94E-02	4.79E-02
	Multiple of Speed (T_0/T_a)		1.92	1.48	1.50	1.39	1.66	2.04	2.08	1.87		1.80
Delta Estimation		Steps	20	80	150	200	250	300	400	500		
	Accel CRR	RMSRE	1.76E-02	4.53E-03	2.23E-03	1.76E-03	1.47E-03	1.09E-03	8.32E-04	7.14E-04		
		Time (T_a)	1.17E-03	1.97E-03	3.09E-03	3.71E-03	4.60E-03	5.39E-03	8.04E-03	9.26E-03		
		Steps	10	20	30	40	50	60	70	80	90	100
	Chen–Joshi	RMSRE	1.53E-02	4.28E-03	2.15E-03	1.56E-03	1.25E-03	1.08E-03	8.93E-04	7.92E-04	7.85E-04	7.64E-04
		Time (T_0)	2.13E-03	4.10E-03	7.19E-03	1.13E-02	1.54E-02	2.07E-02	2.67E-02	3.50E-02	4.58E-02	5.51E-02
	Multiple of Speed (T_0/T_a)		1.82	2.08	2.33	3.04	3.35	3.84	3.32		4.89	

This differential is amplified by an order of 2 to 3 times for delta estimation.

Next, we compare our accelerated CRR model with four analytical formulae briefly alluded to in the literature review: BAW, Bjerksund93, Bjerksund02, and Ju–Zhong. We reuse the 2,330 option parameter sets generated previously. In Exhibits 11 and 12, the results generated from analytical formulae are deliberately placed under the results of Accel CRR that have approximately the same RMSRE. The execution time (Time) with similar levels of accuracy (RMSRE) are investigated. The "Multiple of speed" tentatively maps out how many times our Accel CRR is faster or slower relative to the analytical formulae linked by approximate levels of accuracy. For option pricing (Exhibit 11), Accel CRR is roughly 1.5 times faster than BAW. In contrast, Accel CRR is discernibly slower than Bjerksund93 and Bjerksund02, but almost as fast as Ju–Zhong for similar levels of error. For delta estimation (Exhibit 12), we find that Accel CRR is about 2 and 2.5 times faster than respectively Ju–Zhong and BAW to obtain the same level of accuracy. It is comparable to Bjerksund02, but 0.3 (1 − 0.68) times slower than Bjerksund93. Accel CRR has an obvious advantage in that it provides varying levels of accuracy with different step size. Our accelerated CRR model provides a full spectrum of choice to practitioners varyingly tasked with pricing, repricing,

Exhibit 11
Comparing Accel CRR and Analytical Formulae in Option Pricing

	Steps	10	20	30	40	50	60	70	80	90	100	110	120
Accel CRR	RMSRE	3.56E-02	1.85E-02	1.08E-02	9.98E-03	7.65E-03	6.16E-03	5.58E-03	4.71E-03	4.11E-03	3.61E-03	3.16E-03	2.88E-03
	Time (T_a)	1.10E-03	1.31E-03	1.46E-03	1.62E-03	1.77E-03	1.99E-03	2.08E-03	2.54E-03	2.78E-03	2.91E-03	2.98E-03	3.22E-03
BAW	RMSRE			1.22E-02									
	Time (T_1)			2.24E-03									
Bjerksund93	RMSRE					7.66E-03							
	Time (T_2)					9.91E-04							
Bjerksund02	RMSRE						5.91E-03						
	Time (T_3)						1.38E-03						
Ju–Zhong	RMSRE												2.86E-03
	Time (T_4)												3.10E-03
Multiple of Speed (T_i/T_a, i = 1, 2, 3, 4)				1.53		0.56	0.68						0.96

Exhibit 12
Comparing Accel CRR and Analytical Formulae in Delta Estimation

	Steps	10	20	30	40	50	60	70	80	90	100	120	140
Accel CRR	RMSRE	3.36E-02	1.76E-02	1.04E-02	9.58E-03	7.34E-03	5.99E-03	5.41E-03	4.53E-03	3.99E-03	3.55E-03	2.83E-03	2.40E-03
	Time (T_a)	1.03E-03	1.17E-03	1.36E-03	1.52E-03	1.62E-03	1.74E-03	1.82E-03	1.97E-03	2.04E-03	2.31E-03	2.44E-03	2.67E-03
BAW	RMSRE					7.38E-03							
	Time (T_1)					4.25E-03							
Bjerksund93	RMSRE				7.80E-03								
	Time (T_2)				1.03E-03								
Bjerksund02	RMSRE						7.25E-03						
	Time (T_3)						1.69E-03						
Ju–Zhong	RMSRE											2.61E-03	
	Time (T_4)											4.91E-03	
Multiple of Speed (T_i/T_a, i = 1, 2, 3, 4)					0.68	2.63	1.03					1.92	

and hedging accuracy criteria. Each analytical formula can only provide one combination of speed and accuracy.

In Exhibits 13 and 14, we visualize the performance of our accelerated CRR model, Chen–Joshi, and four analytical formulae for option pricing and delta estimation. Better model performance can be achieved by reaching a closer proximity to the origin. Lower RMSRE and Time indicate higher efficiency. For option pricing (Exhibit 13), Bjerksund93 and Bjerksund02 perform well and efficiency is signaled in terms of their relative proximity to the origin. Accel CRR and Ju–Zhong foster similar levels of error with comparable levels of speed. BAW would appear furthest from the origin, and we interpret this to be consistent with a relatively poorer performance. Accel CRR provides varying combinations of accuracy and speed, represented as a line in the graph, while each analytical formula has only one combination, shown as a single dot. The locus representing Chen–Joshi also contains different levels of accuracy but further from the origin than Accel CRR, which indicates a worse performance. For delta estimation (Exhibit 14), the triangular shape representing Bjerksund93 performs well in terms of proximity to the origin. The other three analytical formulae and Chen–Joshi are farther away from the origin, which means they are less efficient. Again, the locus mapped out by Accel CRR represents different combinations of accuracy and speed relative to the singular dots of analytical formulae. Also, these varying combinations are closer to the origin than Chen–Joshi, which means Accel CRR has relatively better performance.

Exhibit 13

Comparing Accel CRR, Chen–Joshi, and Analytical Formulae in Option Pricing

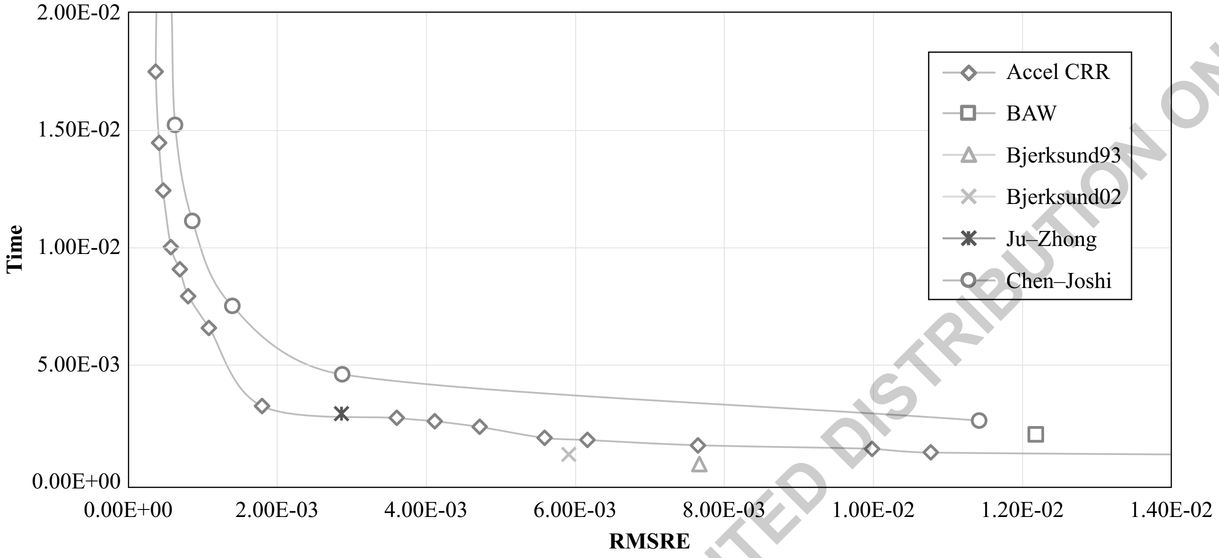
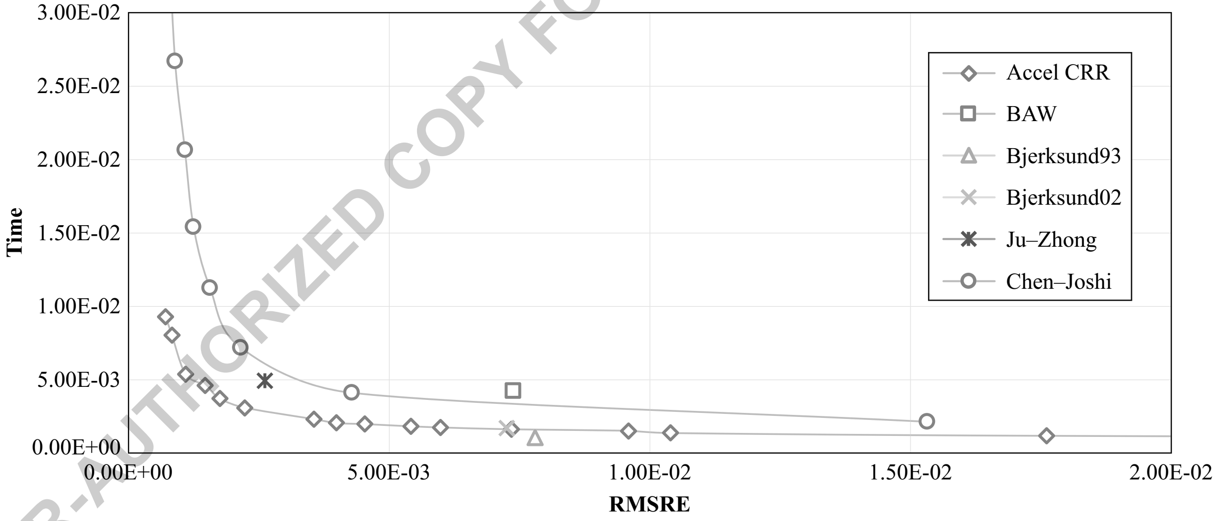


Exhibit 14

Comparing Accel CRR, Chen–Joshi, and Analytical Formulae in Delta Estimation



This final section focuses on the relative efficiency of Accel CRR, Chen–Joshi, and four analytical formulae in estimating implied volatility (IV). We collected chain market prices of all 1,152 live American call and put options on Apple, Inc., from Datastream reported on April 8, 2019, with a contemporaneous stock price of 200.1. The expiry dates span April 18, 2019, May 17, 2019, June 21, 2019, July 19, 2019, August 16, 2019, September 20, 2019, October 18, 2019, January 17, 2020, June 19, 2020, September 18, 2020, January 15, 2021, and June 18, 2021 respectively. The strike price ranges from about 100 to 300 with a uniform interval of 5. The dividend yield is 1.46%, and the risk-free rate, 2.37%.[6] The implied volatility is estimated using the Bisection method, with

[6] The risk-free rate is defined as the annualized three-month US Treasury bill rate on April 30, 2019, obtained from Datastream.

Exhibit 15

Comparing Accel CRR, Chen–Joshi, and Analytical Formulae in Estimating Implied Volatility

	Accel CRR_ 10 Steps	Accel CRR_ 20 Steps	Accel CRR_ 30 Steps	Accel CRR_ 35 Steps	AccelCRR_ 100 Steps	Accel CRR_ 200 Steps	Accel CRR_ 300 Steps	Accel CRR_ 350 Steps
RMSRE	2.99E-02	1.29E-02	9.59E-03	7.71E-03	2.69E-03	1.27E-03	8.51E-04	6.91E-04
Time	3.37E-03	5.41E-03	7.47E-03	8.81E-03	2.38E-02	5.14E-02	8.29E-02	1.01E-01

	Bjerksund93	Bjerksund02	Chen–Joshi_ 10 Steps	BAW	Ju–Zhong	Chen–Joshi_ 20 Steps	Chen–Joshi_ 30 Steps	Chen–Joshi_ 40 Steps	Chen–Joshi_ 50 Steps
RMSRE	4.59E-02	5.78E-02	9.77E-03	8.42E-03	2.14E-03	2.33E-03	1.27E-03	9.24E-04	7.01E-04
Time	1.64E-03	6.02E-03	1.67E-02	4.18E-03	4.83E-03	4.40E-02	8.61E-02	1.41E-01	2.14E-01

a lower and upper bound, respectively, of a = 0.1 and b = 1.[7] The true value of IV is generated using a 15,000-step CRR model combined with Bisection. Three hundred and six of the 1,152 Apple options present with true IV < 0.1% or > 1, or strike price K < 100 or > 300, are excluded. The residual number of options involved in IV estimation is 846 (m = 846). In Exhibit 15, we report the RMSRE and Time of IV estimation using Accel CRR, Chen–Joshi, and four analytical formulae, which is sketched out in Exhibit 16. Again, the sweet spot leans towards lower RMSRE, coupled with lower Time, leading to higher efficiency estimates of IV. In the graph, better model performance can be realized by reaching a closer proximity to the origin. Ju–Zhong represented by the circle is closest to the origin with best performance in IV estimation, followed by BAW, which can achieve the same level of accuracy (RMSRE) with a shorter time than Accel CRR. Bjerksund93 and Bjerksund02 approximately fall on the locus representing Accel CRR, which indicates that they share similar levels of performance. Different from the single data point characteristic of the analytical formulae, Accel CRR provides varying combinations of accuracy (RMSRE) and speed (Time) in IV estimation. Chen–Joshi also possesses this advantage, although it performs worse than Accel CRR, as its locus is further from the origin than Accel CRR.

CONCLUSION

In this article, Cox, Ross, and Rubinstein (1979) is revisited with the ambition to drive down the computational time for estimating American options. The continuous optimal exercise boundary theory proposed by Kim and Byun (1994) and subsequently by Curran (1995) opened a significant vista for reaching greater efficiency for lattice type models despite restrictions being imposed on dividends. We augment their approach to include unrestricted dividends here. Our revamped continuous boundary theory addresses a nontrivial gap for practitioners, as early exercise can occur for an American put (call) option even when r < y (r > y). Furthermore, an intelligent lattice search algorithm is introduced to promptly locate the optimal exercise boundary for American options on assets with unrestricted dividends (yield).

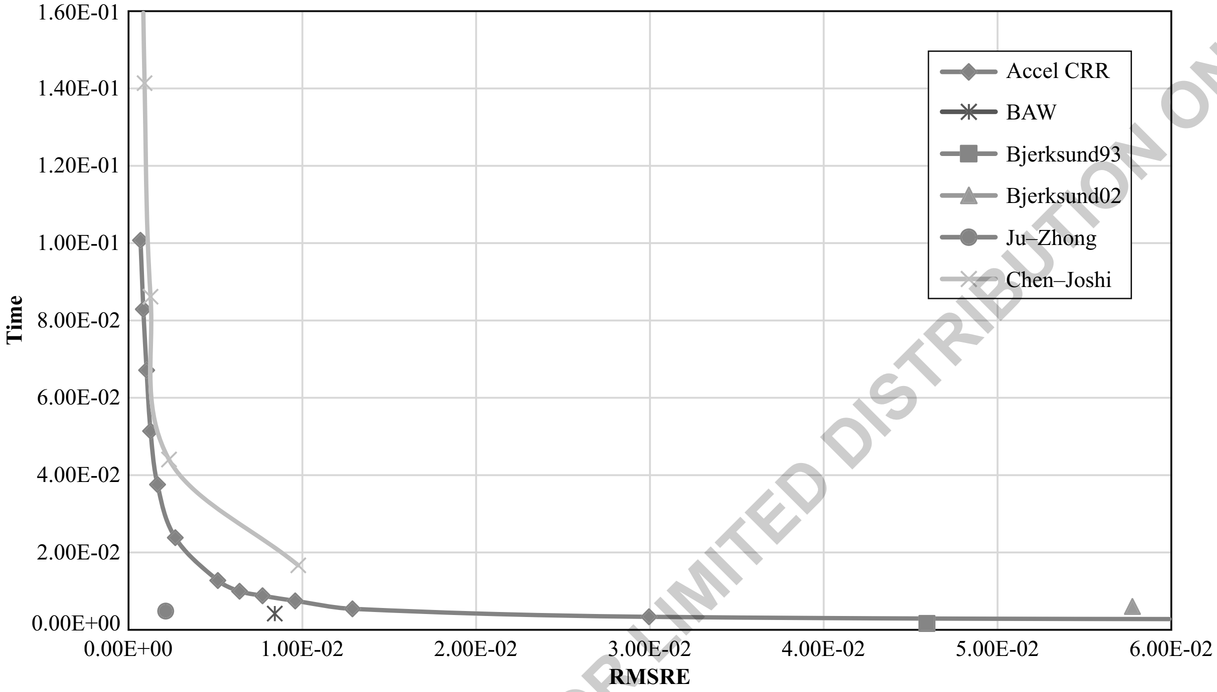
Our accelerated CRR model combines intelligent lattice search, truncation, and dynamic memory technologies.[8] In each instance, we produce equivalent results to the original CRR model. Computational runtime can be reduced from over 18 minutes down to less than 3 seconds to estimate a 15,000-step CRR tree. Significantly, American option pricing and delta estimation are accelerated, in terms of efficacy, dozens of times to hundreds of times, as lattice steps increase. In addition, we compare Accel CRR with the leading benchmark tree, Chen–Joshi, and four popular analytical formulae, including BAW, Bjerksund93, Bjerksund02, and Ju–Zhong. These comparisons are made in terms of option pricing, delta estimation, and implied volatility estimation. Our accelerated CRR model proves to be more efficient than Chen–Joshi and is capable of producing levels of speed consistent with analytical formulae. More importantly, our accelerated CRR model is advantageous to market professionals, in so much as it

[7] Regarding to the Bisection method, please refer to Rouah and Vainberg (2007, p. 9). Since the smallest and largest IV of the residual 846 options are found to reside between 0.18 and 0.98, respectively using a 15,000-step CRR model, we set up the upper and lower bound of 0.1 and 1 correspondingly.

[8] The Excel VBA codes of our accelerated CRR model can be provided upon request.

EXHIBIT 16

Comparing Accel CRR, Chen–Joshi, and Analytical Formulae in Estimating Implied Volatility



flexibly provides varying levels of accuracy with different lattice step size. In contrast, each analytical formula can only afford a single combination of speed and accuracy.

REFERENCES

Areal, N., and A. Rodrigues. 2013. "Fast Trees for Options with Discrete Dividends." *The Journal of Derivatives* 21 (1): 49–63.

Barone-Adesi, G., and R. E. Whaley. 1987. "Efficient Analytic Approximation of American Option Values." *The Journal of Finance* 42 (2): 301–320.

Basso, A., M. Nardon, and P. Pianca. 2002. "Discrete and Continuous Time Approximations of the Optimal Exercise Boundary of American Options." *Quaderni del Dipartimento di Matematica Applicata* 105.

——. 2004. "A Two-Step Simulation Procedure to Analyze the Exercise Features of American Options." *Decisions in Economics and Finance* 27 (1): 35–56.

Bjerksund, P., and G. Stensland. 1993. "Closed-Form Approximation of American Options." *Scandinavian Journal of Management* 9: 87–99.

——. 2002. "Closed-Form Valuation of American Options." *Technical Report*, Norwegian School of Economics and Business Administration, Department of Finance and Management Science.

Broadie, M., and J. Detemple. 1996. "American Option Valuation: New Bounds, Approximations, and a Comparison of Existing Methods." *The Review of Financial Studies* 9 (4): 1211–1250.

Bunch, D. S., and H. Johnson. 1992. "A Simple and Numerically Efficient Valuation Method for American Puts Using a Modified Gesk-Johnson Approach." *The Journal of Finance* 47 (2): 809–816.

Carr, P. 1998. "Randomization and the American Put." *The Review of Financial Studies* 11 (3): 597–626.

Chen, T., and M. Joshi. 2012. "Truncation and Acceleration of the Tian Tree for the Pricing of American Put Options." *Quantitative Finance* 12 (11): 1695–1708.

Cox, J. C., S. A. Ross, and M. Rubinstein. 1979. "Option Pricing: A Simplified Approach." *Journal of Financial Economics* 7 (3): 229–263.

Curran, M. 1995. “Accelerating American Option Pricing in Lattices.” *The Journal of Derivatives* 3 (2): 8–18.

Fabozzi, F. J., T. Paletta, S. Stanescu, and R. Tunaru. 2016. “An Improved Method for Pricing and Hedging Long-Dated American Options.” *European Journal of Operational Research* 254 (2): 656–666.

Figlewski, S., and B. Gao. 1999. “The Adaptive Mesh Model: A New Approach to Efficient Option Pricing.” *Journal of Financial Economics* 53 (3): 313–351.

Geske, R., and H. E. Johnson. 1984. “The American Put Option Valued Analytically.” *The Journal of Finance* 39 (5): 1511–1524.

Haug, E. G. 2007. *The Complete Guide to Option Pricing Formulas*. New York: McGraw-Hill.

Huang, J. Z., M. G. Subrahmanyam, and G. G. Yu. 1996. “Pricing and Hedging American Options: A Recursive Integration Method.” *The Review of Financial Studies* 9 (1): 277–300.

Joshi, M. S. 2009. “The Convergence of Binomial Trees for Pricing the American Put.” *Journal of Risk* 11 (4): 87–108.

Ju, N. 1998. “Pricing by American Option by Approximating its Early Exercise Boundary as a Multipiece Exponential Function.” *The Review of Financial Studies* 11 (3): 627–646.

Ju, N., and R. Zhong. 1999. “An Approximate Formula for Pricing American Options.” *The Journal of Derivatives* 7 (2): 31–40.

Kim, S. J., and I. J. Byun. 1994. “Optimal Exercise Boundary in a Binomial Option Pricing Model.” *Journal of Financial Engineering* 3: 137–158.

Leisen, D. P. J., and M. Reimer. 1996. “Binomial Models for Option Valuation-Examining and Improving Convergence.” *Applied Mathematical Finance* 3 (4): 319–346.

Li, M. 2010. “Analytical Approximations for the Critical Stock Prices of American Options: A Performance Comparison.” *Review of Derivatives Research* 13 (1): 75–99.

McDonald, R. L., and M. D. Schroder. 1998. “A Parity Result for American Options.” *Journal of Computational Finance* 1 (3): 5–13.

Medvedev, A., and O. Scaillet. 2010. “Pricing American Options under Stochastic Volatility and Stochastic Interest Rates.” *Journal of Financial Economics* 98 (1): 145–159.

Rouah, F. D., and G. Vainberg. *Option Pricing Models and Volatility Using Excel-VBA*. Hoboken: John Wiley & Sons. 2007.

Staunton, M. “Efficient Estimates for Valuing American Options.” In *The Best of Wilmott* 2, edited by Paul Wilmott, 91–97. Hoboken: John Wiley & Sons. 2005.

Tian, Y. 1993. “A Modified Lattice Approach to Option Pricing.” *Journal of Futures Markets* 13 (5): 563–577.

Zhu, S. P. 2006. “An Exact and Explicit Solution for the Valuation of American Put Options.” *Quantitative Finance* 6 (3): 229–242.

To order reprints of this article, please contact David Rowe at d.rowe@pageantmedia.com or 646-891-2157.

ADDITIONAL READING

Fast Trees for Options with Discrete Dividends
Nelson Areal and Artur Rodrigues
The Journal of Derivatives
https://jod.pm-research.com/content/21/1/49

ABSTRACT: *The binomial model is a workhorse for valuing options with early exercise possibilities, including options with American or Bermudan exercise and barrier options. The procedure is straightforward when the underlying asset does not pay dividends, and it can be readily adapted to a payout at a fixed proportional rate. But discrete dividends cause trouble because they make the tree "splinter" rather than recombining at each time step. If the tree recombines, an up move followed by a down move produces the same asset price as a down move followed by an up move. But if the intermediate date corresponds to an ex-dividend date and the same fixed amount is subtracted from the up and down nodes, the tree no longer recombines at the subsequent step. The number of nodes in a non-recombining tree increases at a geometric rate, quickly leading to unreasonably long execution times. A number of alternative procedures have been developed to deal with this problem. In this article, Areal and Rodrigues adopt a new approach, accepting the splintering of the binomial lattice but then applying several techniques to accelerate the calculations. They show that their procedures are faster and more accurate than the current methods that force the tree to recombine despite discrete dividend payouts.*

An Approximate Formula for Pricing American Options

Nengjiu Ju and Rui Zhong

The Journal of Derivatives

https://jod.pm-research.com/content/7/2/31

ABSTRACT: *American exercise has always presented a problem for option pricing models. For put options and calls on underlying assets with a continuous proportional cash payout, American exercise turns valuation into a free boundary problem with no closed-form solution. The approximation formula derived by Baroni–Adesi and Whaley has been a very useful tool, but the approximation works best only for short maturities and every long maturities. The formula is less accurate for intermediate maturities of a couple of years, which are now common for over-the-counter option contracts and exchange-traded LEAPS. Other approximation methods based on numerical techniques can be made arbitrarily accurate, but are computationally much more burdensome. Ju and Zhong present a very useful new closed-form model, obtained by introducing correction terms to the Baroni–Adesi and Whaley formula. The model is much more accurate than most alternatives and is also computationally more efficient. As important as improved accuracy, for many users, is the fact that as a closed-form solution, programming Ju and Zhong's model is much simpler than setting up a numerical algorithm.*