On the Enhanced Convergence of Standard Lattice Methods for Option Pricing

MARTIN WIDDICKS
ARI D. ANDRICOPOULOS
DAVID P. NEWTON*
PETER W. DUCK

For derivative securities that must be valued by numerical techniques, the trade-off between accuracy and computation time can be a severe limitation. For standard lattice methods, improvements are achievable by modifying the underlying structure of these lattices; however, convergence usually remains non-monotonic. In an alternative approach of general application, it is shown how to use standard methods, such as Cox, Ross,

The authors would like to thank Syd Howell, Stephen Figlewski, and two anonymous referees for helpful comments on earlier versions of this paper. EPSRC funding is gratefully acknowledged.
*Correspondence author, Manchester Business School, the University of Manchester, Booth Street West, Manchester, M15 6PB, United Kingdom; e-mail: dnewton@man.mbs.ac.uk

Received March 2001; Accepted October 2001

- *Martin Widdicks is in the Department of Mathematics at the University of Manchester in Manchester, United Kingdom.*
- *Ari D. Andricopoulos is in the Department of Mathematics at the University of Manchester in Manchester, United Kingdom.*
- *David P. Newton is at Manchester Business School at the University of Manchester in Manchester, United Kingdom.*
- *Peter W. Duck is in the Department of Mathematics at the University of Manchester in Manchester, United Kingdom.*

The Journal of Futures Markets, Vol. 22, No. 4, 315–338 (2002)
 • DOI: 10.1002/fut. 10010

and Rubinstein (CRR), trinomial trees, or finite differences, to produce uniformly converging numerical results suitable for straightforward extrapolation. The concept of Λ, a normalized distance between the strike price and the node above, is introduced, which has wide ranging significance. Accuracy is improved enormously with computation times reduced, often by orders of magnitude. Jrl Fut Mark 22:315–338, 2002

INTRODUCTION

Binomial and trinomial trees are the most intuitively appealing of the numerical methods for derivatives pricing and are applicable across a wide range of option types. The binomial model of Cox, Ross, and Rubinstein (1979), hereafter referred to as CRR, and the trinomial model of Boyle (1986, 1988) are well known. There have been many adaptations and refinements over the years (for a review, see Broadie & Detemple, 1997).

One of the fundamental problems of a lattice approach is that convergence to the correct option price is oscillatory and non-monotonic. This not only causes the techniques to be inaccurate, but it also renders the data unsuitable for refinement by extrapolation. The well-known non-monotonicity or *sawtooth* effect (see Fig. 1) is due to the misplacement of nodes around the strike price. This article combats the problem by introducing a parameter to describe the position of the strike price with respect to the nodes and illustrates how, by judicious selection of

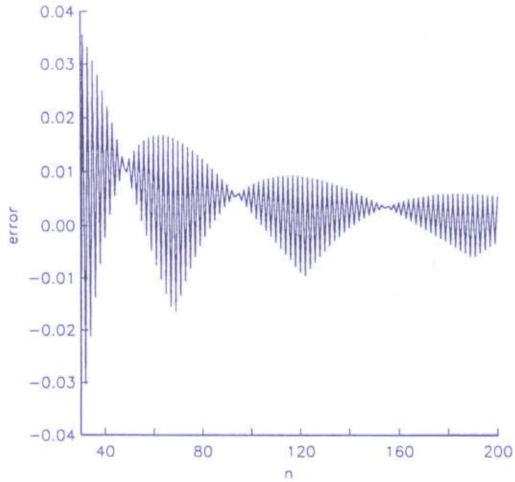

FIGURE 1
A graph of error against n for a European call, using CRR. $S = 100$, $X = 95$, $\sigma = 0.2$, $r = 0.06$, $T = 0.5$.

the number of steps, smooth, monotonic results can be obtained. It then is possible to extrapolate these results to provide a very accurate pricing of a range of options, although the underlying lattice has not been modified in any way.

Continuously traded assets in capital markets are assumed to follow geometric Brownian motion. Immediately invoking the risk-neutrality argument (Harrison & Pliska, 1981), the stochastic process used for derivatives valuation is given by:

$$dS_t = rS_t dt + \sigma S_t dz_t \tag{1}$$

where r is the risk-free rate and σ is the instantaneous standard deviation of the underlying asset $S = (S_t)_{t\geq 0}$ and $(z_t)_{t\geq 0}$ is a standard Gauss–Wiener process, at time t. The problem is transferred into discrete time by constructing a tree to value a derivative with S as the underlying asset.

For the binomial case, consider an asset price S at time $t = 0$. The time to expiry of the derivative, T, is divided into n time steps of equal length, Δt. After a period of time Δt, the price either will have moved up to a new price, Su, or down to Sd, with probabilities p and $(1 - p)$, respectively, where u and d are independent of the asset price. The jump parameters, u and d, and the risk-neutral probabilities are defined according to the variant of the binomial tree chosen and will be specified shortly. If the mean and variance of the binomial distribution equal those of the lognormal distribution, then the binomial lattice method is a discrete time approximation to the continuous time solution. Hence, by Lindeberg's Central Limit Theorem (see Feller, 1971; He, 1990), the binomial method will converge to the Black and Scholes (1973) solution in continuous time. Degrees of freedom remain, allowing different authors to propose alternative definitions for the tree parameters (see Leisen & Reimer, 1996; Tian, 1999). The most common binomial tree for simple applications is CRR, in which, for simplicity, $ud = 1$. The CRR tree also is a benchmark for improved methods. Its parameters are as follows:

$$u = e^{\sigma\sqrt{\Delta t}}, \quad d = \frac{1}{u} \quad p = \frac{e^{r\Delta t} - d}{u - d} \tag{2}$$

Considering the lattice in more detail, the *nodes* S_i at expiry are given by:

$$Su^i d^{n-i}, \quad 0 < i < n, \quad i \text{ integer}, \tag{3}$$

where n is the number of steps in the tree and S is the initial asset price.

Final conditions then are imposed at these nodes depending on which type of option is being valued. An option value at a node one time step earlier can be calculated from the two values, at higher and lower asset prices, to which it is linked in the next time period by discounting at the risk-free rate and weighting according to the risk-neutral probabilities. This is repeated until the option price at time $t = 0$ has been calculated. If option prices at a later time are denoted by v_u and v_d and that at the earlier period by v, then

$$v = e^{-r\Delta t}(p_u v_u + p_d v_d) \tag{4}$$

Adjustments for early exercise are made easily by choosing the maximum of v and the return from an early exercise at each node. Extension to trinomial lattices is in the section entitled Extension to Trinomial Models and is routine.

Figure 1 shows a typical result when using the CRR method to value a European call option. Note the characteristic sawtooth pattern, with high-frequency ringing and low-frequency oscillations (the shrinking envelope or general shape of the plots as the number of time steps in a tree type is increased) due to the position of the nodes at expiry in relation to the strike price. Basic lattice methods are limited by the rate of convergence of this low-frequency envelope; generally, this rate is proportional to $1/n$ as n increases.

By no means all lattice methods neglect the placement of the final nodes. In adapting Ritchken's (1995) model for trinomial lattices, Tian (1999) used an extra degree of freedom in the CRR model to tilt the tree such that one of the nodes at expiry was on the exercise price. This technique (EFB) provides option values corresponding to the values of CRR, which decline nearly monotonically. Extrapolation (which uses two values of n and their respective calculated option values, together with the known rate of convergence to project an expected value at *infinite n*) yields results considerably better than CRR. Unfortunately, convergence is not completely smooth, as necessary changes to the tilt parameter mean that n is not the only variable (i.e., the tilting parameter is itself a complicated and, indeed, unknown function of n). Extrapolation with n ideally requires that all other numerical coefficients be held constant.

Leisen and Reimer (1996) used inversion methods to transform from a lognormal distribution to a binomial one, and set up the lattice so that the strike price was in the center of the final nodes. This scheme is very accurate, but may be regarded as being *tuned* for the European case in that a key element of this scheme involves a priori knowledge of the exact analytic solution for European option prices. Consequently, it does not provide such dramatic improvement in the results for American options (Leisen, 1998), for which the solution only converges at the rate of $1/n$. Also, the highly rigorous nature of the work may be off-putting to practitioners without a strong theoretical background.

Broadie and Detemple (1996) and Gao (1997) avoided the problem of final node placement completely by not calculating them. Instead, the value of the option at the previous timestep was calculated using the analytic Black–Scholes formulae; option valuation then proceeds as normal.

Figlewski and Gao (1999) considered a trinomial lattice and introduced the Adaptive Mesh Model (AMM), in which a finer mesh is incorporated into the structure in the final time step before expiry just around the strike price. The result is described as an AMM1 lattice. The process may be repeated on the fine mesh with still finer meshes, giving an AMMk lattice that provides a marked improvement, but with limited extra effect, as more mesh layers are added, and a basic non-monotonic convergence rate that, as before, improves with $1/n$.

Given the interest, noted by Figlewski and Gao (1999), in finding improvements in speed of calculation to make a series of multidimensional valuations practically feasible, this article proposes a new approach that also will have the appeal of simplicity and will provide greater accuracy for a given computational effort. In the next section, the significance of the final node placement is described, illustrating how and why this is important, and a key parameter, Λ, is defined. A demonstration follows of how, by holding Λ fixed, uniformly converging results are produced and then extrapolated. Results for a range of options then are presented, followed by highlights of the universality of the method with application to trinomial trees.

FINAL NODE PLACEMENT

For options such as European calls, which depend only on conditions at expiry, the positioning of the final nodes significantly affects the accuracy of the results (Leisen & Reimer, 1996). This is because the binomial approximation to the lognormal distribution is dependent on where exactly these final nodes are. The position of the nodes particularly is significant around the strike price, as this is where there is a discontinuity in the delta. A parameter, Λ, will now be introduced that quantifies this positioning.

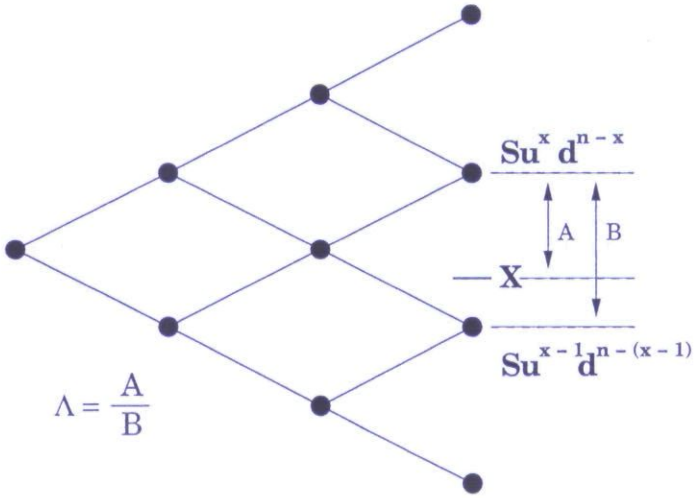

FIGURE 2
A binomial tree indicating how Λ is calculated.

The nodes S_i at expiry are given by equation (3). The distance of each node from the strike price is then given by the absolute value $|S_i - X|$. Suppose that S_x is the closest node greater than the strike price so that

$$S_x - X \leq S_i - X \quad \text{for all } i \quad \text{such that } S_i \geq X \tag{5}$$

If n is increased, the distance between nodes representing different asset prices decreases. Λ is defined in equation (6) as the distance between the strike price and the node at or above it, normalized with respect to the lattice size (see Fig. 2).

$$\Lambda = \frac{S_x - X}{S_x - S_{x-1}} = \frac{Su^x d^{n-x} - X}{Su^x d^{n-x} - Su^{x-1} d^{n-(x-1)}} \tag{6}$$

The concept of a Λ is central to this study and can be applied to binomial, trinomial, and finite difference schemes.

The sensitivity of the solution to the positioning of the final nodes is demonstrated by Figure 3. These simple five- and six-step depictions give an insight into why the binomial CRR lattice gives the results it does. Consider an option with a strike price of \$100. Figure 3 depicts a log-normal distribution of an asset value at $t = T$. The dashed lines correspond to the positions of the nodes and the boxes illustrate how the binomial distribution approximates the normal. If one considers the *envelopes* in Figure 1, then, for a European or American option, the value given by a tree with nodes, as in the left-hand figure, corresponds to that with

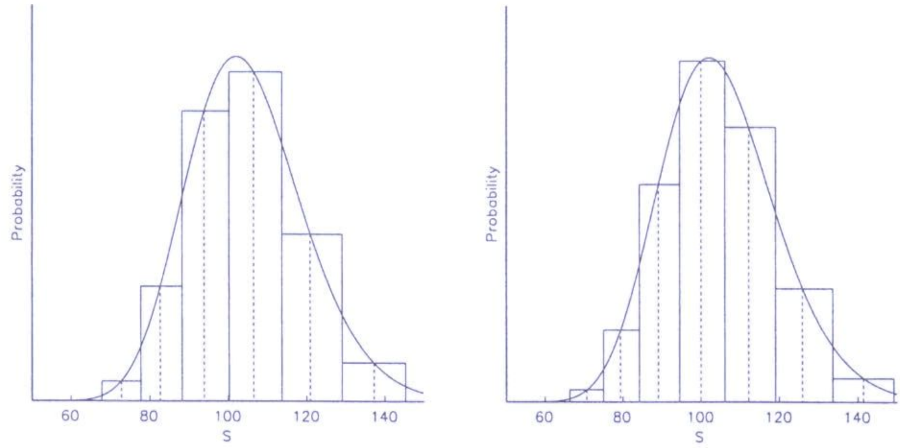

FIGURE 3
The binomial distribution compared to the normal distribution for five- (6 final nodes) and six- (7 final nodes) step trees.

the largest positive error (here, $\Lambda = 0.5$). A tree with nodes, as in the right-hand figure, would produce the largest negative error within the *envelope* (here, $\Lambda = 0$). Note that these two trees are just one step apart.

Thus, the sawtooth pattern is explained by the alternation between the odd and even numbers of nodes and the shape of the envelope by the change in position of the nodes around the strike price as n increases. The decreasing amplitude of the envelopes is due to the improving binomial approximation to the normal. In this context, it makes intuitive sense that if the position of the nodes relative to the strike price could be kept fixed, then convergence would as smooth as the binomial approximation to the normal.

Figure 4 compares the variations of the parameter Λ and the error as n is increased; a precise correlation between the kinks in the latter with jumps in the former is observed. This type of behavior and correlation is generic for tree methods.

When the error is plotted against Λ for a European option, peak-shaped graphs are observed, gradually becoming flatter as n increases. Leisen and Reimer (1996) proved that the convergence of the CRR model for European and American options takes the form $1/n$. If Λ is plotted against $n\times$ error, one would expect the separate distributions to be scaled on top of each other. This is exactly what happens in Figure 5, which depicts a European option; for an American option the picture is similar. The suggestion is there that if Λ could be held constant, then it should be possible to produce smooth monotonic convergence of data.

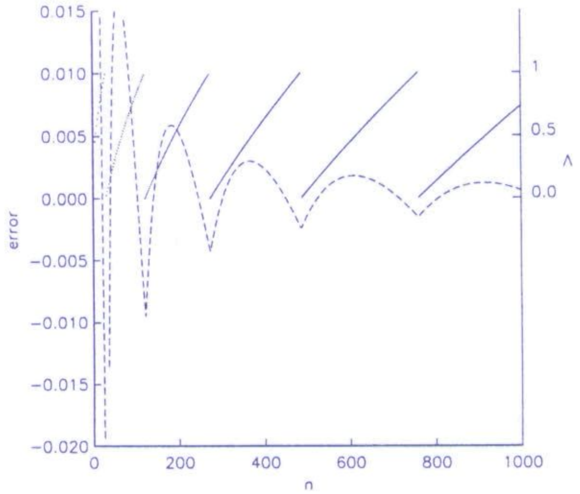

FIGURE 4
A graph of error (for even nodes; dashed), and Λ (dots/solid line), against n for a European call. $S = 100$, $X = 95$, $\sigma = 0.2$, $r = 0.06$, $T = 0.5$.

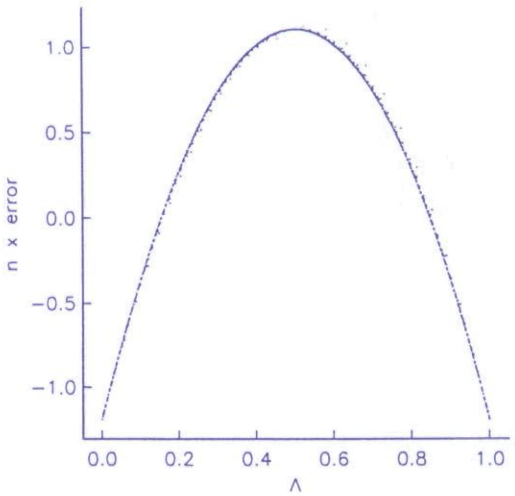

FIGURE 5
A graph of (error $\times n$) against Λ for a European call. $S = 100$, $X = 95$, $\sigma = 0.2$, $r = 0.06$, $T = 0.5$.

CONVERGENCE ACCELERATION OF BINOMIAL LATTICE METHODS

If Λ can be held constant as the number of time steps is altered in a standard tree, such as CRR, then it will be possible to use the valuations from the trees to produce much-improved results in a straightforward fashion.

The constant value of Λ chosen will be 0.5. This is for a number of reasons. First, it is apparent that, for both European and American options, the solution variation with respect to the parameter Λ is smoothest about $\Lambda = 0.5$ (see Fig. 5). In addition, there is a simple technique for finding implied, non-integer values of n in order to maintain this value of Λ and obtain smooth convergence. These advantages outweigh the fact that $\Lambda = 0.5$ generally gives rise to a local maximum error because, as long as there is smooth, monotonic convergence, much of the error can be eradicated through extrapolation. Note that $\Lambda = 0$ or 1 cannot be used because of the interpolation technique employed.

Selecting *n* to Give a Constant Λ for Binomial Methods

In general, there are no two integer values of n for which Λ is exactly the same. Instead, implied non-integer values of n can be taken that correspond precisely to the chosen value of Λ. Conceptually, there is no difficulty with this approach if one considers the inverse of n, which is in effect the timestep Δt. The value of the option at each of these n's may be interpolated from the values of the integer n's on either side using a straightforward second-order scheme. Due to the behavior described in Figure 3 and illustrated in results by the sawtooth effect, there is no continuity between values given by consecutive n's. Smooth results, as in Figure 4, come when either all odd or all even values of n are chosen. Thus, either both n's must be odd or both n's must be even. If the chosen Λ is termed Λ_{fixed} and the value of Λ for n steps is termed Λ_n, then m and $m + 2$ must be found such that

$$\Lambda_m \le \Lambda_{\text{fixed}}, \quad \text{and} \quad \Lambda_{m+2} \ge \Lambda_{\text{fixed}} \quad \text{for } S_{t=0} < X \tag{7}$$

or,

$$\Lambda_m \ge \Lambda_{\text{fixed}}, \quad \text{and} \quad \Lambda_{m+2} \le \Lambda_{\text{fixed}} \quad \text{for } S_{t=0} > X \tag{8}$$

One way of achieving this would be to search through the final nodes on every tree, but this can be time consuming and there is a more efficient method. For the case of $\Lambda_{\text{fixed}} = 0.5$, the following procedure finds the values of the implied n's (n_{imp}) such that a node one time step before expiry lands on the strike price. This gives a Λ of approximately 0.5, which, though not exact, is sufficient for practical purposes (noting that $\Lambda \to 0.5$ as $n \to \infty$). Furthermore, this value usefully provides an accurate starting guess for a Newton–Raphson iteration process that can be implemented to yield the exact value of n. The final nodes of a CRR tree with n steps, initial price S, can be written as:

$$Su^{n-i}d^{i} = Su^{n-2i}\left(\text{as } d = \frac{1}{u}\right), \quad \text{where } i = 1, 2, \ldots, n \tag{9}$$

The penultimate nodes can be written as:

$$Su^{n-2i-1} \quad \text{where } i = 1, 2, \ldots, n-1 \tag{10}$$

Let I denote the integer value $n - 2i - 1$. Note that if I is odd, then n must be even and vice versa. The requirement that a penultimate node lies exactly on the strike price, X, can be written as:

$$Su^{I} = X \tag{11}$$

In a CRR tree, the number of time steps implied is found easily:

$$u = e^{\sigma\sqrt{\Delta t}} = e^{(\sigma\sqrt{T}/\sqrt{n_{\text{imp}}})} \tag{12}$$

and so

$$n_{\text{imp}} = \left(\frac{I\sigma\sqrt{T}}{\log(\frac{X}{S})}\right)^2, \quad \text{where for } X > S, \quad I = 2, 4, 6, \ldots \text{ or } I = 3, 5, 7, \ldots \tag{13}$$

If $S > X$, then I will be negative (i.e., $I = -2, -4, -6, \ldots$ or $-3, -5, -7, \ldots$).

To further improve the estimate for n_{imp} with fixed Λ, the following equation must be satisfied:

$$\frac{Su^{I+1} - X}{Su^{I+1} - Su^{I-1}} = \Lambda_{\text{fixed}} \tag{14}$$

or

$$(I-1)\log(u) + \log((1-\Lambda_{\text{fixed}})u^2 + \Lambda_{\text{fixed}}) - \log\left(\frac{X}{S}\right) = 0 \tag{15}$$

Simple Newton–Raphson iteration of this equation with equation (12) quickly leads to an improved/exact value for n_{imp} with $\Lambda = 0.5$ (or indeed any other value) and is very easy to program (Press, Teukolsky, Vetterling, & Flannery, 1992). In practice, with $\Lambda = 0.5$, the values of n_{imp} from equation (13) serve as excellent starting guesses for the solution of equations (14) or (15).[1]

Next, the option value implied for the non-integer number of step, n_{imp}, can be calculated. Option values from two real trees, with integer values of n, are needed. If even values of I are taken, then trees must be produced with $N = m$ and $N = m + 2$, where m is the nearest odd integer below n_{imp}. If odd values of I are taken, then m must be the nearest even integer below n_{imp}. The option values given by these lattices will be termed V_m and V_{m+2}. For each implied n, the implied option value, V_{imp}, is given by:

$$
\begin{aligned}
V_{\text{imp}} &= V_m + \frac{(V_{m+2} - V_m)(n_{\text{imp}} - m)}{m + 2 - m} \\
&= V_m + \frac{(V_{m+2} - V_m)(n_{\text{imp}} - m)}{2} \qquad (16)
\end{aligned}
$$

We now have a method for generating values, V_{imp}, suitable for extrapolation, as will be shown in the next section.

Note that the case $S = X$ is somewhat simpler, and by choosing either even or odd values, smooth monotonic and uniform convergence with rate $1/n$ may be obtained. The case is simpler because any odd n gives rise to $\Lambda = 0.5$; as a result, there is no need to calculate n_{imp} or V_{imp}, as these are m (odd) and V_m. Then it is possible to proceed directly to extrapolation.

Extrapolation

Smooth monotonic results now have been achieved. In the chosen case, with $\Lambda = 0.5$, they do correspond to the locally highest possible values (the high points of the sawtooth pattern) given by the CRR lattice (for

[1]An anonymous referee kindly pointed out that using such a root finding technique would not be necessary if one were to use log-price space (lnS) as opposed to price space (S). This is because the equivalent equation to 15 is a linear equation in n_{imp} for each I. Such a transformation may save a small amount of time; however, the length of time it takes to perform the Newton–Raphson iteration is insignificant in comparison to the time taken to calculate the four binomial trees required in the full method. Hence, a log-transformation, although perhaps more elegant, makes no discernible difference to the overall efficiency of the method. The referee also noted that, in Figure 5, the smoothest point is, in fact, where $\Lambda = 0.5$ on the equivalent tree in ln(S) space. In S space, this is equivalent to a Λ slightly greater than 0.5. The difference in results seen in practice, however, is negligible.

European and American options), but that will be seen to be inconsequential compared with the benefits from extrapolation. However, unlike the values selected by Tian (1999), the data are perfectly smooth and monotonic.

Let V_E be the exact price of the option, and V_n be the value given by CRR after n steps. The rate of convergence as $n \to \infty$ for European and American options for a fixed Λ is known to be $1/n$; therefore, V_n is assumed to take the following form:

$$V_{n_{\mathrm{imp}}} = V_E + \frac{f_1(\Lambda)}{n_{\mathrm{imp}}} + \text{smaller terms} \tag{17}$$

where f_1 is an unknown function of Λ. This suggests an extrapolation technique to improve the accuracy of the model by removing the $1/n$ term. If two implied values of n with the same value of Λ (n_1 and n_2, for example) are taken and the implied option values (V_{n_1} and V_{n_2}) are calculated, then a better approximation to V_E can be found. Ignoring the smaller terms in equation (17):

$$f_1(\Lambda) = n_1 V_{n_1} + n_1 V_E = n_2 V_{n_2} + n_2 V_E \tag{18}$$

which can be rearranged to give:

$$V_E = \frac{n_1 V_{n_1} - n_2 V_{n_2}}{n_1 - n_2} \tag{19}$$

The key point to note here is that f_1 is dependent only on Λ and not on n. Note that $V_{n_{\mathrm{imp}}}$ could refer not just to the option value, but also to the implied value of one of the *Greek* parameters, which are calculated in the next section. Finally, if the discontinuities (e.g., strike price or barrier) are positioned correctly, then all tree methods converge with $1/n$.

Calculating the Greeks

Practitioners familiar with calculating the hedge parameters would have found that the sawtooth pattern exhibited by the valuation of options using the binomial method is mirrored when valuing the Greeks. The parameters, Delta, Gamma, Rho, Vega, and Theta are all affected by Λ in a similar manner to the option price. To improve the accuracy of these valuations, simply calculate the hedge parameters as normal for each binomial lattice, find the interpolated value, and extrapolate in exactly the same way as for the option price. Note that Rho and Vega require two lattices, each with different r or σ, respectively, and, in order that the shape of the lattice to not be affected significantly, the difference between the two values used should be kept to a minimum. Results, as with the option valuation, are very impressive; for an indication of these, see Table II in the next section.

Implied Volatilities

The present method lends itself perfectly to calculation of implied volatilities using Newton–Raphson iteration. The integer parameter I should be kept fixed and Newton–Raphson can be used to find accurately the σ such that $V_{\text{real}} - V_{\sigma} = 0$ (where V_{real} is the actual option price and V_{σ} is that given by the present method with fixed I and volatility σ). Results are once again a considerable improvement on CRR; an example is shown in Table II.

RESULTS

The tables and graphs in the sub-sections that follow compare the results for the present method (denoted by WAND) against other methods. CRR is Cox, Ross, and Rubinstein's (1979) original binomial method. EFB is Tian's (1999) Extended Flexible Binomial Model and BBSR is Broadie and Detemple's (1996) quasi-analytic binomial approach extrapolated once. LEIS are Leisen and Reimer's model for European (1996) and Leisen's for American (1998) options. Trinomial models, namely the basic trinomial model (TRIN) and the Adaptive Mesh Model (AMMk) of Figlewski and Gao (1999), will be considered in the next section.

Note that EFB uses two (tilted) trees, as does LEIS (1998), whilst WAND uses four standard CRR trees (two are required to get the implied value at n_{imp} and two are required for the extrapolation). The results shown count the number of steps as being the number of steps of the largest tree used in the extrapolation. Newton–Raphson iteration was used to calculate the implied n's from equation (14). For WAND, the extrapolation was performed with n_{imp} and the previous n_{imp} calculated from I and $I - 2$ (or $I + 2$ for $S < X$), respectively.

For each case, a range of options are considered and the root of mean squared errors (RMSE) with respect to the analytic result, or the previous best benchmark in the case of American puts (i.e., LEIS method with $n > 50000$), are calculated for a given number of steps in the tree. These options are based around an initial asset price of 40 and risk-free rate of 0.06. The other parameters considered are strike prices of 35 and 45, volatilities of 0.2 and 0.4, and maturities of 0.25 and 0.5.

In the table, there is an entry WAND n, which is because it is not possible to designate exact values of n for this method. Values close to the desired n were selected for each option and the average taken. These were the same for European calls, Cash-or-Nothing calls, and American puts.

To give an indication of relative performances, run times on a computer also are given. Timings are given just in the case of European calls, as they are approximately the same for the other types of options, the exception being LEIS for the American puts, which has a similar run time to that of EFB. These were calculated on a modest desktop computer (Pentium III-550). Programs were written in Fortran and compiled using the NAG Fortran 95 v4.0 compiler.

European Options

Although the method is not restricted to options for which analytical solutions are available, comparison with such solutions for European options permits an accurate demonstration of numerical trends. The results are excellent. Figures 6 and 7 compare the current method with CRR and EFB, respectively. The unextrapolated V_{imp} values are shown by

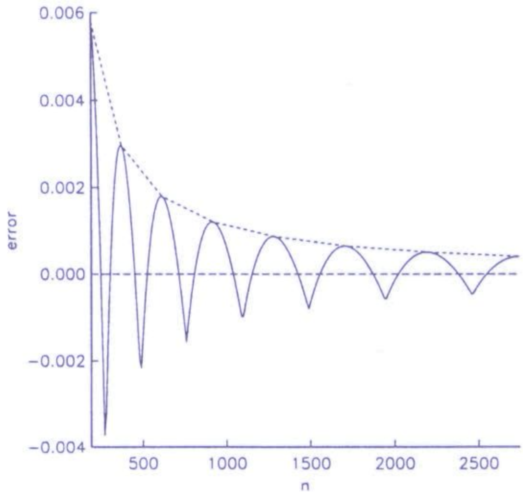

FIGURE 6
A graph of error against n for a European call. CRR, even steps (solid line), unextrapolated (small dashes), and WAND extrapolated (thicker dashes). $S = 100$, $X = 95$, $\sigma = 0.2$, $r = 0.06$, $T = 0.5$.

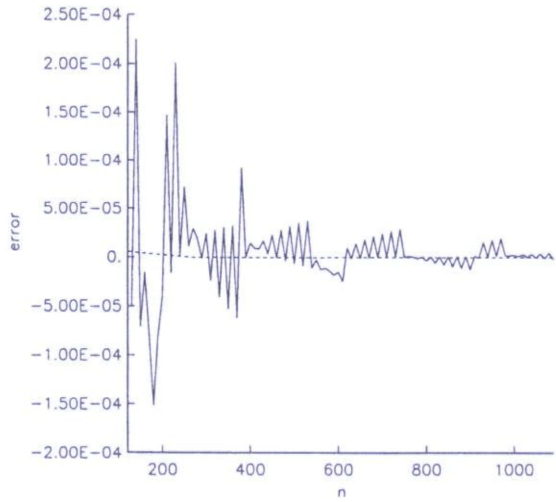

FIGURE 7
A graph of error against n for a European call. Extended Flexible Binomial, every 10 steps (solid line) and WAND extrapolated (dashed). $S = 100$, $X = 95$, $\sigma = 0.2$, $r = 0.06$, $T = 0.5$.

the dashed line passing through the peaks of the CRR scheme in Figure 6.

Table I shows how the method introduced in this study (WAND) outperforms basic tree and adjusted-parameter models, represented by the CRR, EFB, and BBSR methods, as well as being at least comparable with results from Leisen and Reimer's more sophisticated *tuned* model (1996) over this broad range of options. In Table I, after 0.12 seconds, WAND gives an RMS error of 3×10^{-7}. It is not shown on the table, but even after 50,000 steps, taking nearly 7 min, CRR still gives an RMS error of 8×10^{-6}. The potential for use in more demanding applications is clear.

By plotting results for one example on a graph of $\log_{10}(n)$ against $\log_{10}(\text{error})$ (see Fig. 8), it can be established that the present method converges with order approximately n^2, compared with the unextrapolated CRR results, which converge with order n.

It now will be shown that this technique can be used to improve numerical convergence in the pricing of other classes of option.

Cash-or-Nothing Options

The generality of the approach can be demonstrated for options with discontinuous payoffs. Cash-or-Nothing options (abbreviated to CONs) are a variation on European options in that if, at expiry, the option is in the

TABLE I

A Comparison of the Performance of the Current Method, WAND, Against Other Methods Over a Range of Option Types and Parameters

n (*WAND n*)	*WAND Error* (*time/s*)	*CRR Error* (*time/s*)	*EFB Error* (*time/s*)	*LEIS Error* (*time/s*)	*BBSR Error* (*time/s*)
Approximation RMS Error for European Calls					
50	0.000111832	0.009955261	0.005345188	0.000066772	0.000133687
(51)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
100	0.000022457	0.003964939	0.000785480	0.000017204	0.000110792
(98)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
500	0.000000304	0.000865978	0.000136878	0.000000705	0.000006807
(511)	(0.12)	(0.03)	(0.04)	(0.04)	(0.05)
1000	0.000000387	0.000498266	0.000039393	0.000000177	0.000004793
(1008)	(0.47)	(0.14)	(0.18)	(0.15)	(0.18)
5000	0.000000033	0.000068617	0.000004419	0.000000007	0.000001022
(4946)	(13.08)	(3.66)	(4.58)	(3.72)	(4.52)
Approximation RMS Error for CON Calls					
50	0.000039589	0.022406985			
100	0.000012851	0.018469219			
500	0.000000268	0.004712786			
1000	0.000000145	0.003703826			
5000	0.000000011	0.001622794			
Approximation RMS Error for American Puts					
50	0.001597782	0.009188979	0.003389657	0.035913428	0.002095184
100	0.000241921	0.002971765	0.000906200	0.000468642	0.000634463
500	0.000024700	0.000890818	0.000078560	0.000027412	0.000048742
1000	0.000012157	0.000515600	0.000039824	0.000018501	0.000025340
5000	0.000000366	0.000077265	0.000002783	0.000000554	0.000003320

Note. RMS errors and run times averaged over the eight different sets of parameters are displayed. CRR, EFB, LEIS, and BBSR models are as described in the text. WAND *n* is the average of the steps used in each of these eight cases for the WAND technique. The models are based upon $S = 40$ and $r = 0.06$, no dividends and, for the Cash-or-Nothings, $q = 1$. The option test set for each of the three types of option consists of combinations of strike prices of 35 and 45; maturities of 0.25 and 0.5, and volatilities of 0.2 and 0.4.

TABLE II

A Comparison of RMS Error for the *Greeks* and Implied Volatility Using WAND and CRR Methods

Parameter	*n* (*WAND n*)	*WAND Error*	*CRR Error*
Approximation RMS Error for the Greeks for European Calls			
Delta	500 (511)	0.00000017	0.00009826
Gamma	500 (511)	0.00000005	0.00001962
Vega ($\delta\sigma = 0.001$)	500 (513)	0.00184476	0.09750915
Rho ($\delta_r = 0.0001$)	500 (511)	0.00000277	0.00151223
Approximation RMS Error for the Implied Volatility for American Puts			
Implied volatility	500 (511)	0.00000450	0.00011797

Note. RMS errors are calculated for the same set of options as in Table I. WAND *n* is as defined earlier.

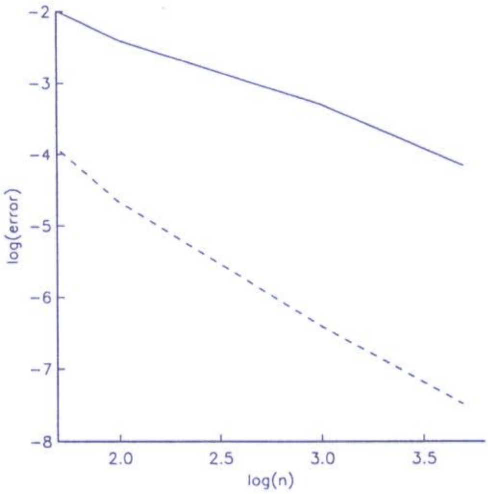

FIGURE 8
A graph of $\log_{10}$(RMS error) against $\log_{10}(n)$ for European call options for the CRR method (solid line) and WAND extrapolated (dashed line). For the 8 cases described in Table I with $n \approx 50$, 100, 1000, 5000.

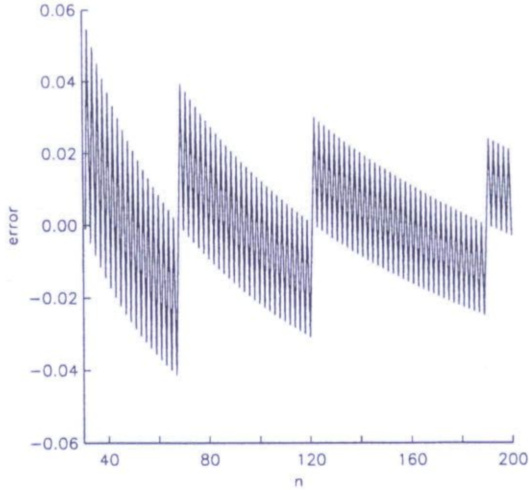

FIGURE 9
A graph of error against n for a Cash-or-Nothing call, using CRR. $S = 100$, $X = 95$, $\sigma = 0.2$, $r = 0.06$, $T = 0.5$, $q = 1$.

money, then the amount received is at a set level q. Consequently, at expiry, the payoff is either 0 or q depending on the price of the underlying asset. Figure 9 shows a typical result for pricing these options using the CRR method.

Although, with CRR, the convergence of CONs to the analytic solution is at a slower rate, $1/\sqrt{n}$, they actually are far simpler options than their European counterparts. The value of the option is basically the area under the lognormal distribution of S at $t = T$ on the appropriate side of the strike price, multiplied by q, and discounted back at expiry.

However, there are now two components of the error term that need to be removed. The smaller component, which decays as $1/n$, is the difference between the binomial approximation to the lognormal and the lognormal itself over the whole graph. The larger component, which decays as $1/\sqrt{n}$, is produced by the misplacing of the nodes around the strike price. If Λ is not equal to 0.5, then it is equivalent to putting the strike price at the point where Λ is equal to 0.5. As the standard deviation of the normal varies with $1/\sqrt{n}$, then so must this miscalculated area. Incidentally, the reason why this is not the case for a European option is that the different payouts for different values of $S_{t=T}$ compensate for the misplacement of the nodes.

For the case of CONs, then, (17) must be replaced by

$$V_n = V_E + \frac{f_0(\Lambda)}{n^{1/2}} + \frac{f_1(\Lambda)}{n} + \text{smaller terms} \tag{20}$$

where $f_0(\Lambda)$ and $f_1(\Lambda)$ are both independent of n. However, taking $\Lambda = 0.5$ gives $f_0 = 0$, and then it is possible to use precisely the same procedure as for the European options, with extrapolation based on just two values of V_{imp}. The results obtained using the method shown particularly are impressive. Plotting the log–log graph for Cash-or-Nothing options (not shown) demonstrates that, as with European options, the present method converges with order n^2. Table I shows clearly the remarkable improvement made on CRR by the present method.

American Put Options

For both of the previous option classes, exercise was permitted only at expiry. Thus, it is largely at expiry that the position of the nodes is important. American options are considerably more complex in that there is a possibility of exercising early and, hence, there is a condition at each time step, implying that all the nodes in the lattice are important. It is this free boundary that has prevented an analytic, closed-form solution from being found. The question now is whether, by using precisely the same method as for European options, an improvement can be made in the evaluation of American options.

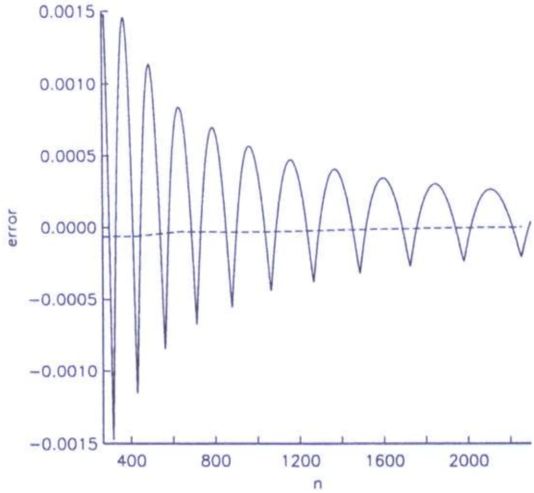

FIGURE 10
A graph of error against n for an American put. CRR, even steps (solid line) and WAND extrapolated (dashed). $S = 100$, $X = 110$, $\sigma = 0.2$, $r = 0.06$, $T = 0.5$.

A comparison with CRR is shown in Figure 10. Plotting the log–log graphs (not shown) indicates that although the coefficient of convergence is better than CRR, the convergence is still only with n, as is the unextrapolated scheme. Leisen (1998) had the same problem and, to date, no basic lattice scheme with fixed time steps has overcome this difficulty, although recently Leisen (1999) has presented a random-time binomial model that suggests $1/n^2$ convergence.

In the standard lattice approach, the procedure is to check at each node whether or not early exercise would be optimal. This gives rise to a discrete approximation to the continuous free boundary, and the asymptotic form as $n \to \infty$ is not as conclusive as for European (or indeed CONs). The likely explanation of this is that f_1 in (17) is not truly a function only of the parameter Λ, but rather will depend (in a nontrivial way) on the location of the free boundary. This includes dependence upon where and how this boundary intersects with the lattice, thereby introducing many additional lengthscales into the computation. Thus, the quantity f_1 will be a complicated (possibly even pseudo-random) function of the grid. However, the overwhelming evidence is that the function f_1 remains $O(1)$, thereby ensuring the $1/n$ convergence of the solution, albeit in a nonuniform fashion. Despite the underlying difficulties, the results continue to be very impressive and provide the most accurate results of all the models considered for a given number of steps.

EXTENSION TO TRINOMIAL MODELS

The use of a fixed value of the key parameter Λ to improve the accuracy and convergence of CRR methods has been successful for valuing European, Cash-or-Nothing, and American options. It remains to be shown that it is feasible to extend this procedure for use on trinomial schemes because the associated parameters also may be chosen irrespective of final node placement.

The trinomial model used here is a very basic one, with $m = u \times d = 1$. The following choices of u, d, p_u, p_m, and p_d are taken:

$$u = e^{\sigma\sqrt{3\Delta t}}, \qquad d = \frac{1}{u} \tag{21}$$

$$p_u = \sqrt{\frac{\Delta t}{12\sigma^2}}\left(r - \frac{\sigma^2}{2}\right) + \frac{1}{6}, \quad p_m = \frac{2}{3}, \quad p_d = -\sqrt{\frac{\Delta t}{12\sigma^2}}\left(r - \frac{\sigma^2}{2}\right) + \frac{1}{6} \tag{22}$$

This is probably the most widely used choice of parameters and indicates how the present method can be used in conjunction with trinomial trees, as long as $u \times d = 1$ and the parameters are not selected so as to fix the final node placement. A plot of $n\times$ error against Λ for European and American options gives rise to a graph almost identical to that in Figure 5 for binomial trees. Hence, an extrapolation technique again can be used.

In a similar way to the binomial method, let the ith node at expiry be S_i, with asset price given by Su^{n-i} (where $0 \leq i \leq 2n$). The implied non-integer value of n, which gives rise to the specified value of Λ, may be calculated as in the binomial case, but without the problems associated with odd and even steps. This is because when the number of steps is increased by one, all the previous nodes practically are preserved and two new ones added. This is easy to see by considering one step of a trinomial to be equivalent to two steps of a binomial tree.

Whereas for binomial trees, there is a straightforward procedure to calculate a good approximation to the implied n for $\Lambda = 0.5$; this cannot be done for the trinomial tree. For the binomial case, the appropriate estimate was given by a node before expiry landing on the strike price. Using the trinomial lattice, if there is a node on the strike price at the time step before expiry, there still will be one there at expiry, and, hence, $\Lambda = 0$.

However, it is possible to achieve an estimate for the implied n by searching through the final nodes on every tree ($n = 2, 3, 4, \ldots$) in order to find the closest node above the strike price and, thus, Λ. Then, using

TABLE III

A Comparison of RMS Errors When Pricing European Call Options With Various Trinomial Models

n (WANDt n)	*WANDt Error (time/s)*	*TRIN Error (time/s)*	*AMM4 Error (time/s)*	*AMM12 Error (time/s)*
Approximation RMS Error for European Calls				
50 (65)	0.000029693 (0.00)	0.005048763 (0.00)	0.000015861 (0.00)	0.000030795 (0.00)
100 (100)	0.000011822 (0.01)	0.002238289 (0.00)	0.000026547 (0.00)	0.000020260 (0.00)
500 (493)	0.000000463 (0.28)	0.000523528 (0.09)	0.000003998 (0.08)	0.000003992 (0.09)
1000 (999)	0.000000119 (1.21)	0.000221272 (0.34)	0.000001744 (0.35)	0.000001955 (0.36)
5000 (4967)	0.000000006 (33.21)	0.000096643 (8.88)	0.000000534 (9.08)	0.000000423 (9.08)

Note. WANDt, TRIN, and AMMk are as defined in the current section. RMS errors and timings were calculated for the same set of option parameters described in Table I. WAND *n* is as defined earlier.

the integer values m and $m + 1$, either side of the desired Λ interpolation can be used to find n_{imp}. This procedure is sufficient for practical purposes, however, to find the implied values of n more precisely and efficiently, Newton–Raphson iteration was used based on the formula

$$\frac{Su^{I} - X}{Su^{I} - Su^{I-1}} - \Lambda_{\text{fixed}} = 0 \tag{23}$$

where $u = e^{\sigma\sqrt{3(T-t)/n_{\text{imp}}}}$, $I = 1, 2, 3, \ldots$ for $S < X$ and $I = -1, -2, -3, \ldots$ for $S > X$. Note that when $X = S$, any value of n can be used.

The results displayed in Table III indicate a marked improvement of the present method—denoted WANDt—on other trinomial schemes for the base case of European call options. This trend would extend to the other options considered for the binomial case although the binomial WAND is more accurate than the trinomial at pricing American puts.

In terms of node calculations, WANDt is considerably more efficient than other trinomial models despite the need for four trees. Figure 11 displays total node calculations in comparison to AMM4. Note that although AMM4 is considerably better than AMMk ($k < 4$), refining the method further to AMMk ($k > 4$) does not continue to produce significant improvement. By comparison of run times, the importance of this becomes apparent. An RMS error of 5×10^{-7} is obtained using WANDt after an average 493 steps and 0.3 s. AMM4 achieves the same error after 5000 steps, and it takes approximately 9 s. Overall, the extension to a trinomial lattice has shown the flexibility of the concept of Λ. The results, however, are actually no better than WAND results for

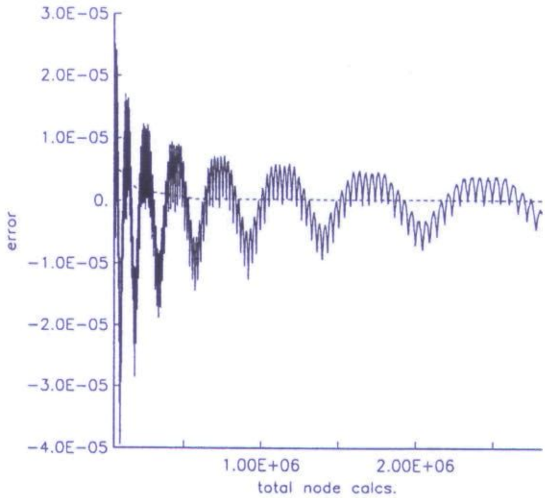

FIGURE 11
A graph of error against total calculations for AMM4 (solid line) and WANDt (dashed) for a European Call. $S = 100$, $X = 110$, $\sigma = 0.2$, $r = 0.06$, $T = 0.5$.

binomial lattices, and they do require more calculations. Thus, the binomial method is more efficient than the trinomial.

CONCLUSIONS

In this study, the well-known oscillations of the binomial model, due largely to the position of the nodes at expiry in relation to the strike price, have been further investigated, using as examples European, Cash-or-Nothing, and American options. By introducing the concept of Λ, a technique has been found that uses results from unmodified standard trees to produce regularly converging data suitable for use in an extrapolation technique, which then leads to startlingly accurate results. The method has been extended for use with trinomial trees to demonstrate that it is a universal technique for any lattice scheme that does not explicitly fix the final nodes, and also for finite difference methods that do. The method is seen consistently to outperform a number of alternatives (for example, Figlewski & Gao, 1999; Tian, 1999). It is simple to understand and implement, requiring only results from standard trees plus straightforward manipulation of those results. The extension of this method to problems in two or more dimensions could lead to tremendous reductions in computational time.

APPENDIX

Pseudo code

```
Lambda = 0.5
j = 0
DO INT  = 4, 6, 2
     j = j + 1
     IF (X < S) INT = -INT
     IF (X = S) THEN
        M = 25*INT
        GOTO 100
     END IF
     eta = (sig*sqrt(t)*INT/log(X/S))**2
     CALL NEWTON(eta,S,X,sig,t,Lambda)
     N = NINT(eta) + 1
     IF ( N-1 < eta) N = N + 1
     CALL ODDEVEN(N,even)
     IF (even = 1) N = N-1
     M = N-2
     CALL CRR(N,S,X,sig,r,t,Vn)
100  CALL CRR(M,S,X,sig,r,t,Vm)
     Nimp(j) = eta
     V(j) = (Vn-Vm)*(eta - M)/(N-M) + Vm
     IF ( X = S ) THEN
        Nimp(j) = M
        V(j) = Vm
     END IF
     IF(j>1) CALL EXTRAP(Nimp(j),V(j),Nimp(J-1),V(j-1),Vext (j))
END DO
```

In the above pseudo code, the subroutine *NEWTON* takes the estimated value of *eta* and, using Newton–Raphson iteration (as described previously), improves it; the subroutine *ODDEVEN* checks whether *N* is even (if so, *even* = 1); *CRR* is simply a CRR binomial tree with output *Vn* or *Vm* (this can be either the value of the option or one of the *Greeks*); *EXTRAP* is the extrapolation procedure detailed earlier outputting the value of *Vext*.

BIBLIOGRAPHY

Black, F., & Scholes, M. (1973). The pricing of options and corporate liabilities. Journal of Political Economy, 81, 637–659.

Boyle, P. P. (1986). Option valuation using a three-jump process. International Options Journal, 3, 7–12.

Boyle, P. P. (1988). A lattice framework for option pricing with two state variables. Journal of Financial and Quantitative Analysis, 35, 1–12.

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

Broadie, M., & Detemple, J. (1997). Recent advances in numerical methods for pricing derivative securities. In L. C. G. Rogers & D. Talay (Eds.), Numerical methods in finance (pp. 17–43). Cambridge, UK: Cambridge University Press.

Cox, J. C., Ross, S. A., & Rubinstein, M. (1979). Option pricing: a simplified approach. Journal of Financial Economics, 7, 229–264.

Feller, W. (1971). An introduction to probability theory and its applications (2nd ed.). New York: Wiley.

Figlewski, S., & Gao, B. (1999). The adaptive mesh model: a new approach to efficient option pricing. Journal of Financial Economics, 53, 313–351.

Gao, B. (1997). Convergence rate of option prices from discrete- to continuous-time (working paper). University of North Carolina.

Harrison, J., & Pliska, S. (1981). Martingales and stochastic integrals in the theory of continuous trading. Stochastic Processes and Their Application, 11, 215–260.

He, H. (1990). Convergence from discrete- to continuous-time contingent claims prices. Review of Financial Studies, 3, 523–546.

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

Leisen, D. P. J. (1999). The random-time binomial model. Journal of Economic Dynamics & Control, 23, 1355–1386.

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

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (1992). Numerical recipes in Fortran, the art of scientific computing (2nd ed.). Cambridge, UK: Cambridge University Press.

Ritchken, P. (1995). On pricing barrier options. Journal of Derivatives, 3, 19–28.

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