Curtailing the Range for Lattice and Grid Methods

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

ARI D. ANDRICOPOULOS graduated from the doctoral program in mathematics at the University of Manchester, Manchester, UK.

MARTIN WIDDICKS is a lecturer in finance at the Manchester Business School at the University of Manchester, Manchester, UK.

PETER W. DUCK is a professor of mathematics at the University of Manchester, Manchester, UK.

DAVID P. NEWTON is a senior lecturer in finance and director of the Executive MBA Program at the Manchester Business School at the University of Manchester, Manchester, UK.

Valuing options using standard lattice and finite-difference methods often involves much extra computational effort because the grid or lattice generally includes computations at levels of the underlying that are practically unfeasible. This article details a method of general application that curtails ranges of the underlying for which calculations are carried out, achieving considerable savings in computational effort with virtually no loss in accuracy. Extremely simple to implement and wide-ranging in scope, this is a canonical tool that can be used to complement virtually any grid or lattice method.

The application of finite-difference or lattice methods can entail a significant amount of superfluous computational effort. The probability that the logarithm of an underlying asset will move more than a certain number of standard deviations from its starting value is so low that it can be ignored. Similarly, when there is one underlying asset, if the value of the underlying at maturity is known, the probability that the asset path will move outside that number of standard deviations in working backward is just as low.

In other words, for certain values of the underlying, the strike price is comparatively so low or so high that it could not realistically be reached, and for these values the option price is effectively deterministic. This idea is demonstrated in Exhibit 1, which illustrates how many nodal calculations are superfluous, assuming a feasible range of ξ standard deviations, in valuing a plain vanilla American or European put option on one underlying asset with a binomial tree.

By calculating only the values of nodes within the feasible range of the underlying, S, it is possible to save a great deal of computational effort without significant loss of accuracy. This method is applicable to a great variety of options. We classify it here as curtailed range binomials (CR BIN) and curtailed range finite-differences (CR FD).

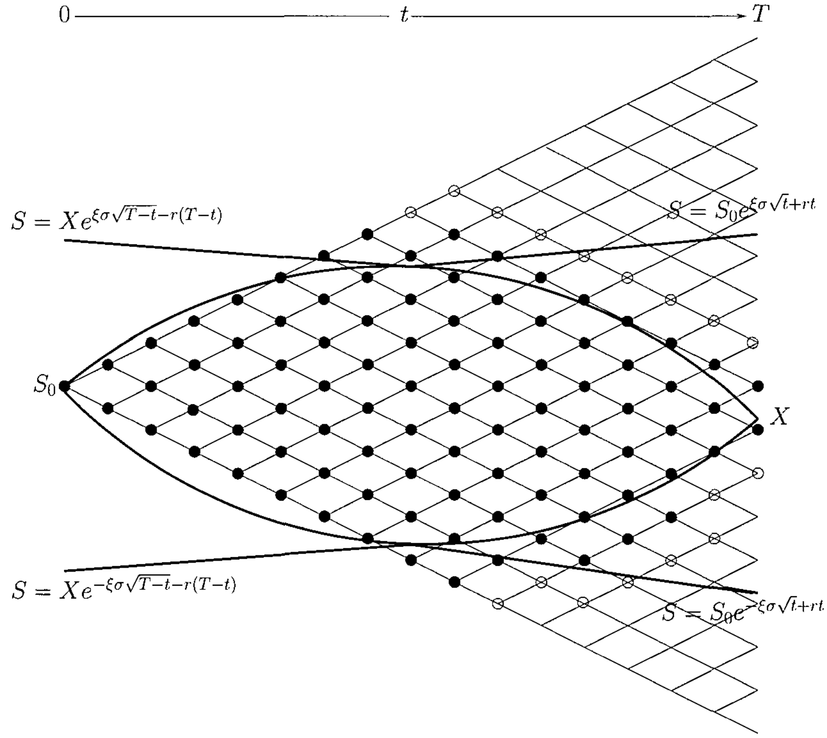
The idea can be extended to multiple dimensions, where even greater improvements in efficiency are possible, although the two-directional approach is not applicable for more than one underlying. Actual savings in computational time vary depending upon the efficiency of the lattice or finite-difference grid to which the method is applied, but would generally be at least 40%, for reasonably large trees.

I. METHOD FOR ONE UNDERLYING ASSET

We demonstrate the method for a binomial tree evaluating an American put option. Generalization from this CR BIN method to CR FD is routine.

The first stage is to define the range of S in which the option values must be calculated, as shown in Exhibit 1. If the option runs from time 0 to time T ($T > 0$), and a line of nodes (in S) is at t ($0 < t < T$), the maximum and minimum required values of S_{max} and S_{min} are:

EXHIBIT 1
Curtailed Range



The solid circles represent values that must be evaluated using the values at the next time step; the open circles represent values that can be set without using the next values. All other nodes can be ignored.

$$S_{max} = \min\left(S_0e^{rt+\xi\sigma\sqrt{t}}, Xe^{-r(T-t)+\xi\sigma\sqrt{T-t}}\right)$$
$$S_{min} = \max\left(S_0e^{rt-\xi\sigma\sqrt{t}}, Xe^{-r(T-t)-\xi\sigma\sqrt{T-t}}\right) \tag{1}$$

where ξ determines the extent of the range the user wishes to incorporate and is the number of standard deviations from the expected value. In practice, ξ of between 5.0 and 7.5 should be used, depending on the level of accuracy required.

Above S_{max}, the value of the put option should be taken to be zero as either it is unfeasible that the underlying will reach that value, or unfeasible that at that value the underlying can then return below the strike price, or both. In the case of the American put option, below S_{min}, the value of the option should be taken as $X - S$, as the probability that the underlying will reach that value and then go above the early exercise boundary is too low to have any effect on the option value. If the option is a European put, below S_{min} the value would be set at $Xe^{-r(T-t)} - S$ for similar reasons.

Note that if the option value is an unknown function of S below S_{min} (as, for example, when a discrete down-and-out barrier is imposed on the put option), then for obvious reasons the curtailment is such that:

$$S_{min} = S_0e^{rt-\xi\sigma\sqrt{t}} \tag{2}$$

as it is only possible to curtail moving forward in time for S_{min}. There is a parallel when the option value is an unknown function of S above S_{max}, such as when valuing an up-and-out discrete barrier call.

The method is divided into three simple stages that are repeated N times, from $t = T$ working backward in time:

- *Stage 1:* At each time step (t), using either the usual binomial tree method or, at maturity, the known exercise conditions, evaluate the option at all the nodes that lie within the range $[S_{min}, S_{max}]$ as well as at one node outside the range on either side to ensure that the range is fully covered.

EXHIBIT 2
Efficiency Comparisons—One Underlying

Valuing an American put option using a binomial tree

N	BIN (time (s))	CR BIN (time (s))	Difference in value (%) (Difference in time (%))
500	2.5192500	2.5192500	0.00000
	(0.03)	(0.01)	(-67)
1000	2.5207757	2.5207757	0.00000
	(0.15)	(0.02)	(-87)
5,000	2.5201193	2.5201193	0.00000
	(4.01)	(0.25)	(-94)
10,000	2.5200521	2.5200521	0.00000
	(16.86)	(0.69)	(-96)

Valuing a down-and-out discrete barrier call option using a finite-difference method

N	FD (time (s))	CR FD (time (s))	Difference in value (%) (Difference in time (%))
1000	9.42089	9.42085	-0.00042
	(0.05)	(0.03)	(-40)
2000	9.41731	9.41722	-0.00096
	(0.15)	(0.08)	(-47)
5000	9.41526	9.41520	-0.00063
	(0.54)	(0.28)	(-48)
10,000	9.41459	9.41452	-0.00074
	(1.51)	(0.81)	(-46)
20,000	9.41423	9.41411	-0.00127
	(4.62)	(2.13)	(-54)

American put option parameters: $S = 100$, $X = 95$, $T - t = 0.5$, $\sigma = 0.2$, $r = 0.06$, and $\xi = 6$.

Discrete barrier option parameters: $S = 100$, $X = 95$, $BAR = 90$, $M = 125$, $T - t = 0.5$, $\sigma = 0.2$, $r = 0.06$, and $\xi = 5$ where M is the number of times the barrier was observed. Run times using 1.4 Ghz AMD Athlon processor.

- *Stage 2:* Find the S_{max} and S_{min} at the previous time step (which is $t - \Delta t$) and the range of nodes at which the option must be evaluated at this time step.
- *Stage 3:* Note that to evaluate the option at the range of nodes calculated in Stage 2 (at time $t - \Delta t$), it may be necessary to use values at time t that have not been evaluated in Stage 1. The values at these nodes are known a priori, however, i.e., either 0 above S_{max} or $X - S$ below S_{min} (for a European put, this would be $Xe^{-r(T-t)} - S$ below S_{min}). These should be set as such. The three stages should then be repeated, at $t - \Delta t$, $t - 2\Delta t$, until time 0 is reached.

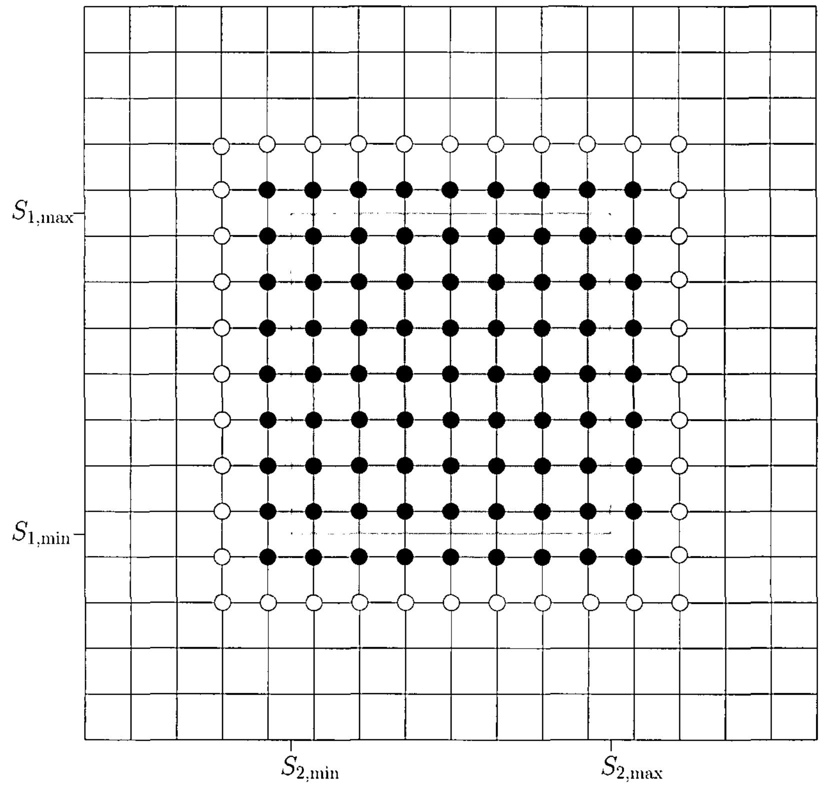
The simplest way to demonstrate the technique is in a pseudo-code, as given in the appendix, where we annotate the three stages for an American put option using the Cox, Ross, and Rubinstein [1979] binomial method (CRR).

Exhibit 2 shows comparisons between raw computations for a binomial tree (American put option) and a finite-difference grid (down-and-out call). The former uses the CRR approach and the latter the Crank-Nicolson [1947] method. As would be expected, as the tree size grows, overall reductions become increasingly dramatic (a 96% reduction for a 10,000-step tree).

As a better indicator in more challenging situations (although in this case still straightforward to implement), the method reduces computation time by around 50% in pricing a discrete down-and-out barrier call option using the Crank-Nicolson method.

EXHIBIT 3

Curtailed Range at One Time for Lattice on Two Underlying Assets



The shaded region denotes the curtailed region. The solid circles denote the values that need to be calculated at this time. The open circles denote the values that may be required at the next time step (working backward) but that can be set without performing any calculations.

II. METHOD FOR MULTIPLE ASSETS AND COMPLEX PATH-DEPENDENT OPTIONS

The basic methodology can be extended to improve standard lattice and grid methods when there are more underlying assets. The same is true for valuing options with complex path-dependent features that require another dimension, i.e., lookback or Asian options.[1] Unfortunately, due to the more complex topography of the payoff regions, it is generally not possible to curtail working backward from expiration, but only to curtail working forward from $t = 0$.

To demonstrate the curtailed range technique for multiasset problems, consider an American put option on two underlying assets, S_1 and S_2 with a payoff $\max(X_1 - S_1, X_2 - S_2, 0)$. Curtailing only forward in time, $S_{i,min}$ and $S_{i,max}$ are given by:

$$S_{i,\min} = S_{i,0}e^{rt-\xi\sigma_i\sqrt{t}}$$

$$S_{i,\max} = S_{i,0}e^{rt+\xi\sigma_i\sqrt{t}}$$

for $i = 1, 2$, so at each time t there will be a rectangular region of values to calculate in the region $S_{1,\min} \leq S_1 \leq S_{1,\max}$, $S_{2,\min} \leq S_2 \leq S_{2,\max}$.

The method is best illustrated by use of a diagram as in Exhibit 3. Here the shaded region represents the curtailed range in both S_1 and S_2 at time t. The nodes that need to be calculated in Stage 1 are denoted by the solid circles and are evaluated using the option values at $t + \Delta t$.

To perform the valuation at the next time step ($t - \Delta t$), the values denoted by the open circles may be required —see Stage 2. Stage 2, in fact, can be ignored when the curtailment is applied only forward in time, because no more than one layer of extra values will ever be required. These are not calculated by using the lattice or grid but are input as deterministic values—this is Stage 3. In this case these are $\max(X_1 - S_1, X_2 - S_2, 0)$.[2]

Extension to options on more than two underlyings follows a similar procedure, although it can become a little more difficult to visualize the curtailed region and hence set the deterministic values required in Stage 3.

Exhibit 4
Efficiency Comparison—Multiple Underlyings

Valuing an American put option on two underlying assets using a binomial tree: payoff $= \max(X_1 - S_1, X_2 - S_2, 0)$

N	BEG (time (s))	CR BEG (time (s))	Difference in value (%) (Difference in time (%))
100	4.2841024	4.2841024	0.00000
	(0.03)	(0.01)	(-66)
200	4.2862727	4.2862727	0.00000
	(0.25)	(0.06)	(-76)
500	4.2789369	4.2789369	0.00000
	(5.26)	(0.47)	(-91)
1000	4.2812302	4.2812302	0.00000
	(44.46)	(1.91)	(-96)
2000	4.2803356	4.2803355	0.00000
	(377.81)	(8.37)	(-98)

Valuing an American put option on three underlying assets using a binomial tree: payoff $= \max(X_1 - S_1, X_2 - S_2, X_3 - S_3, 0)$

N	BEG (time (s))	CR BEG (time (s))	Difference in value (%) (Difference in time (%))
50	5.6233827	5.6233827	0.00000
	(0.16)	(0.07)	(-56)
100	5.6185345	5.6185342	0.00000
	(7.50)	(0.66)	(-91)
200	5.6225494	5.6225482	0.00000
	(129.27)	(6.94)	(-95)
300	5.6161763	5.6161739	0.00000
	(658.23)	(23.20)	(-96)
400	5.6199033	5.6198995	0.00000
	(2119.06)	(51.05)	(-98)
500	5.6155429	5.6155375	0.00000
	(5247.56)	(91.60)	(-98)

Parameters: $S_i = 100$, $X_i = 95$, $T - t = 0.5$, $\sigma_i = 0.2$, $\rho_{ij} = 0.25$, $r = 0.06$, and $\xi = 5$. Run times using a 2.5 Ghz Pentium IV processor.

Exhibit 4 shows comparisons between raw computations for a binomial tree calculating the value of an American put option on two and three underlying assets. The binomial tree calculation is based on the Boyle, Evnine, and Gibbs [1989] method, using the generalization by Kamrad and Ritchken [1991] to k underlying assets (where $k \geq 2$).

The method is as follows. For a lattice with k underlying assets, the probabilities, p_m, for each of the $m(= 2^k)$ possible states at the next time step are given by:

$$p_m = \frac{1}{2^k}\{1 + \sqrt{\frac{(T-t)}{N}}\sum_{i=1}^{k} x_{im}\frac{r - \frac{1}{2}\sigma_i^2}{\sigma_i} + \sum_{i=1}^{k-1}\sum_{j=i+1}^{k} x_{ij}^{m}\rho_{ij}\} \quad (3)$$

where N is the number of time steps in the lattice;

$$x_{im} = \begin{cases} 1 & \text{if asset } i \text{ has an up jump in state m} \\ -1 & \text{if asset } i \text{ has a down jump in state m} \end{cases} \quad (4)$$

$$x_{ij}^{m} = \begin{cases} 1 & \text{if assets i and j have jumps in the same direction in state m} \\ -1 & \text{if assets i and j have jumps in the opposite direction in state m} \end{cases} \quad (5)$$

Even though the curtailment is possible in only one direction when there is more than one underlying asset, the curtailed range method remains effective. As in the case of one underlying asset, as the tree size grows, overall reductions in computations become increasingly dramatic. We can achieve, for example, a 98% improvement for a 2,000-step tree with two underlying assets and a 500-step tree with three underlying assets.

III. CONCLUSION

The curtailed range method is a useful, simple, and neat extra tool in numerical option pricing. It makes it easy to achieve improvements in calculations, and it can be incorporated into a wide variety of techniques for more complex cases.

APPENDIX

Pseudo-Code

A pseudo-code demonstrates how to implement the curtailed range method to price an American put option on one underlying asset using the CRR binomial method.

```
ARRAY V, ip, im
REAL S, X, t, r, sigma, u, d, p, delt, s1, Smax, Smin,
   Smax2, Smin2, xi
INTEGER i,j

READ S, X, t, sigma, r, N, xi !read in the parameters, N
   and xi.

delt = t/N
u = exp(sigma*sqrt(delt))
d = 1/u
p = (exp(r*delt) - d)/(u - d) !the CRR u, d and p

!The values are now set for the final time step:

Smax = X*exp(- r*delt + xi*sqrt(delt)*sigma)
Smin = X*exp(- r*delt - xi*sqrt(delt)*sigma)

ip(1) = 0.5*(N-log(Smax/S)/log(u))
im(1) = 0.5*(N-log(Smin/S)/log(u)) + 1

!the reason 1 is added to the im(1) is that integers are automatically
!rounded down, so this ensures that a node either side of the range
is calculated.

DO i  = ip(1), im(1) + 1 !these are required at time (T –
   delt)
   S1  = S*u**(N – 2*i)
   V(i) = X – S1
   IF (V(i) < = 0) THEN
      V(i) = 0
   ENDIF
ENDDO

!Now for all earlier time steps:
DO j = 1, N

!Stage 1:

DO i = ip(j), im(j)
   S1 = s*u**(N – j – 2*i)
   V(i) = exp(-r*delt)*(p*V(i) + (1 – p)*V(i + 1))
   IF(V(i) < (X – S1)) V(i) = X – S1
ENDDO

!Stage 2:

IF(j < N-1) THEN
   Smax = exp(log(X) -r*(j + 1)*delt + rg*sqrt((j + 1)*delt)*
    sigma)
   Smin = exp(log(X) -r*(j+1)*delt - rg*sqrt((j + 1)*delt)*
    sigma)
   Smax2 = exp(log(S) +r*(N-j-1)*delt +rg*sqrt((N-j-1)*
    delt)*sigma)
   Smin2 = exp(log(S) +r*(N-j-1)*delt -rg*sqrt((N-j-1)*
    delt)*sigma)
IF(Smax > Smax2) Smax = Smax2
IF(Smin < Smin2) Smin = Smin2
END IF !this IF stopped sqrt(0) occurring.
ip(j+1) = 0.5*(N-j-1-log(Smax/S)/log(u))
im(j+1) = 0.5*(N-j-1-log(Smin/S)/log(u)) + 1

!Stage 3:

DO i = ip(j + 1), ip (j)
    V(i) = 0
ENDDO

DO i = im(j + 1) + 1, im(j),-1
   S1 = S*u**(-j-2*i)
   V(i) = X-S1
ENDDO

ENDDO
PRINT 'value = ', V(0)
```

ENDNOTES

The authors gratefully acknowledge funding from the EPSRC and the Leverhulme Trust.

[1]To value path-dependent options curtailment is possible only working forward in time due to the dependence of the payoff on both the stochastic variables. If the other dimension is, for example, stochastic volatility, it would be possible to curtail in both directions, as demonstrated for one underlying asset, because in this case the payoff is dependent only on the value of the underlying asset, not the other stochastic variable, the volatility.

[2]For the sake of simplicity, the range in Exhibit 3 is assumed to be rectangular. If one takes into account the joint distribution of the two variables, it is possible to improve the efficiency of the method still further. The corners of the rectangular region represent extremely rare events whose probabilities are the product of two (small) univariate probabilities, so the region can be curtailed further. Given this, the most efficient region would be elliptical, taking into account the correlation between the two variables. Although this may increase computational complexity, it may well also provide a more efficient curtailed range method.

REFERENCES

Boyle, P.P., J. Evnine, and S. Gibbs. "Numerical Evaluation of Multivariate Contingent Claims." *Review of Financial Studies*, 2 (1989), pp. 241-250.

Cox, J., S.A. Ross, and M. Rubinstein. "Option Pricing: A Simplified Approach." *Journal of Financial Economics*, 7 (1979), pp. 229-264.

Crank, J., and P. Nicolson. "A Practical Method for Numerical Evaluation of Solutions of Partial Differential Equations of the Heat Conduction Type." *Proceedings of the Cambridge Philosophical Society*, 43 (1947), pp. 50-67.

Kamrad, B., and P. Ritchken. "Multinomial Approximating Models with *k* State Variables." *Management Science*, 37 (1991), pp. 1640-1652.