Fast, accurate and inelegant valuation of American options

Adriaan Joubert & L.C.G. Rogers [1]

In this short note, we describe the pricing of an American option (call or put) on a share which may pay continuous dividends, using a method described by Broadie & Detemple [2] as 'not very elegant'. The method is simply to build a look-up table of option prices, which thus splits the problem of pricing American option prices into three sub-problems:

(A) accurate calculation of the values in the table;
(B) storage of the table, and access to it;
(C) rapid calculation of prices for given parameter values.

The most important problems are clearly B and C; in principle, we may take as long as necessary to fill up the table, since this calculation is done off-line, once only. Any of the methods discussed elsewhere in this volume by Broadie & Detemple [2] and by Ait-Sahalia & Carr [1] could be used to compute the values in the table. We used the binomial method with 5000 time steps using Black-Scholes in the last step, as recommended by Broadie & Detemple.

It is worth remarking that by computing and storing the table of values, we are able to calculate greeks, and the exercise boundary with relatively little extra cost; this is a valuable advantage of this inelegant approach.

To describe the storage problem, let us first state the parametrisation which we used. The price of an American put option written on a share paying continuous dividends at rate δ is

$$P(S_0, K, r, \sigma, T, \delta) \equiv \sup_{0 \le \tau \le T} E[(Ke^{-r\tau} - S_0 e^{\sigma W(\tau) - (\delta + \sigma^2/2)\tau})^+],$$

where T is the expiry of the option, τ is a stopping time with values in $[0, T]$, K is the strike price, S_0 is the price of the share at time 0, and σ is the volatility of the share returns. Though the price ostensibly depends on six parameters, we can reduce the problem somewhat by rewriting

$$\begin{aligned} P(S_0, K, r, \sigma, T, \delta) &= \sup_{0 \le \tau \le \sigma^2 T} E[(Ke^{-r\sigma^{-2}\tau} - S_0 e^{W(\tau) - (\delta\sigma^{-2} + (1/2))\tau})^+] \\ &\equiv P(S_0, K, r\sigma^{-2}, 1, \sigma^2 T, \delta\sigma^{-2}) \\ &= KP(S_0/K, 1, r\sigma^{-2}, 1, \sigma^2 T, \delta\sigma^{-2}) \\ &\equiv Kp(S_0/K, rT, \sigma^2 T, \delta T), \end{aligned}$$

say; thus there are effectively only four parameters of the problem, S_0/K, $\sigma^2 T$, rT, and δT. We therefore only have to store a four-dimensional table, which is feasible. In fact,

[1]Published in the Proceedings of the Numerical Methods Workshop at the Isaac Newton Institute, April 1995, eds L C G Rogers and D Talay, Cambridge University Press, 1997

the ranges we took were:

S/K	[0.7 , 1.3]	T	[0.003, 1.0]
r	[0.01,0.10]	σ	[0.03 ,0.60]
δ	[0.00,0.10]		

There are many practical difficulties in the construction of such a look-up table. Close to the exercise boundary the option pricing function p changes very rapidly, and at the exercise boundary its second derivative is large. Interpolating in intervals that straddle the exercise boundary causes larger errors. For this reason the exercise boundary is pinpointed accurately for the S/K grid lines while the table is constructed, and its position stored in a separate table. Furthermore S/K is confined to values where options have a significant value, i.e. are bigger than 10^{-5}. Consequently the grid spacing decreases when the option has a short running time — this is essential to obtain reasonable estimates for short maturities.

The grid spacing for δT is fixed, but for every grid point in the δT dimension bounds for rT are determined, so that only the range of parameters that are of interest are covered. The reason is that choosing δT, and having a fixed range for δ imposes restrictions on the values for T that have to be considered. The same is true for the grid for $\sigma\sqrt{T}$. This simple optimisation reduces the size of the table roughly by half.

We also experimented with non-equidistant grids in order to have more points in regions where T is assumed to be small, but the slight improvement in precision did not seem to justify the additional complication in accessing the table and interpolating the option prices.

For the results presented here, a table with 21 points in the T and S/K dimensions and 16 points in the δ and σ dimensions is used. The total amount of space required to store this was 0.94 MB, which is quite acceptable (it easily fits on a single floppy disc).

It is possible to use so little storage only because we use judicious choice of grids and a polynomial interpolation method. There are various interpolation schemes well known in numerical analysis; we used a modified Neville interpolation. For an account of the method, see, for example, the book by Stoer [3].

For the interpolation 4 points are used in every grid dimension, giving a total of 256 points. Heuristics are use to exclude some points if they are outside the exercise boundary. If the option price is not close to the exercise boundary and has a long maturity, quite acceptable results can be achieved using only quadratic or even linear interpolation. In the results reported here the interpolation is not optimised to choose the number of interpolation points dynamically.

Table displays results for short term puts with a strike price of $100 and no dividends. Here S is the stock price, T the expiry of the option, σ the volatility and r the risk-free interest rate. The results in the 'Binomial' column were calculated with a 5000-step

S ($)	T (Years)	σ (%)	r (%)	Binomial ($)	Look-Up ($)	Rel. Err.	Abs. Err.
80	0.500	40	6	21.6057	21.6057	0.0000	0.0000
85	0.500	40	6	18.0370	18.0368	0.0000	0.0002
90	0.500	40	6	14.9178	14.9178	0.0000	0.0000
95	0.500	40	6	12.2306	12.2303	0.0000	0.0003
100	0.500	40	6	9.9448	9.9454	-0.0001	-0.0006
105	0.500	40	6	8.0265	8.0267	0.0000	-0.0003
110	0.500	40	6	6.4337	6.4339	0.0000	-0.0002
115	0.500	40	6	5.1257	5.1253	0.0001	0.0003
120	0.500	40	6	4.0602	4.0601	0.0000	0.0001
100	0.500	40	2	10.7734	10.7739	0.0000	-0.0005
100	0.500	40	4	10.3441	10.3447	-0.0001	-0.0006
100	0.500	40	6	9.9448	9.9454	-0.0001	-0.0006
100	0.500	40	8	9.5707	9.5712	-0.0001	-0.0005
100	0.500	40	10	9.2186	9.2191	0.0000	-0.0005
100	0.500	30	6	7.2110	7.2114	-0.0001	-0.0004
100	0.500	35	6	8.5774	8.5779	-0.0001	-0.0005
100	0.500	40	6	9.9448	9.9454	-0.0001	-0.0006
100	0.500	45	6	11.3116	11.3123	-0.0001	-0.0006
100	0.500	50	6	12.6767	12.6774	-0.0001	-0.0007
100	0.083	40	6	4.3732	4.3743	-0.0003	-0.0012
100	0.167	40	6	6.0716	6.0718	0.0000	-0.0002
100	0.250	40	6	7.3070	7.3074	-0.0001	-0.0004
100	0.333	40	6	8.3149	8.3154	-0.0001	-0.0005
100	0.417	40	6	9.1869	9.1874	-0.0001	-0.0005
100	0.500	40	6	9.9448	9.9454	-0.0001	-0.0006
100	0.583	40	6	10.6247	10.6253	-0.0001	-0.0006
100	0.667	40	6	11.2500	11.2506	-0.0001	-0.0006
100	0.750	40	6	11.8172	11.8178	-0.0001	-0.0006
100	0.833	40	6	12.3424	12.3430	-0.0001	-0.0006
100	0.917	40	6	12.8374	12.8380	-0.0001	-0.0007

Seconds Per Put : 0.00027
Mean Relative Error $\left(\frac{1}{n}\sum \epsilon_i\right)$: -0.000045
Root mean square error $\left(\sqrt{\frac{1}{n}\sum \epsilon_i^2}\right)$: 0.000068
Mean Absolute Relative Error $\left(\frac{1}{n}\sum |\epsilon_i|\right)$: 0.000053
Maximum Relative Error ($\max |\epsilon_i|$) : -0.000263
Maximum Absolute Error ($\max |p_i - \hat{p}_i|$) : 0.001150
(Here $n = 30$, p_i=Binomial price, $\hat{p}_i$=Lookup Table price, $\epsilon_i = \frac{p_i - \hat{p}_i}{p_i}$)

binomial tree, using Black-Scholes in the last step. The relative and absolute errors are given in the last two columns.

Even in a first implementation, this method can price around 3000 options per second with high accuracy. We used a similar test to Broadie & Detemple for 5000 options, with parameters chosen independently from a uniform distribution, such that:

K	100	S	[70 ,130]
T	[0.1, 1.0]	r	[0.01,0.10]
σ	[0.1,0.60]	δ	[0.00,0.10]

We follow them in rejecting options with a price less then 0.005, which left 4687 option prices; we similarly define dhe relative error as $\epsilon = (p - \hat{p})/p$, where p is the put price calculated with a 5000-step binomial algorithm, using Black-Scholes in the last step, and $\hat{p}$ is the price obtained from the look-up table. The relative root-mean-square error (RMS) is then defined as

$$\sqrt{\frac{1}{n}\sum\left(\frac{p-\hat{p}}{p}\right)^2}\ .$$

With the look-up table defined above, the relative root-mean-square error was 0.001001, and the maximum relative error (chosen such that 99.5% of the observations have smaller errors) was -0.006011. Warnings are issued for options for which the enclosing cube in the $(\delta T, rT, \sigma\sqrt{T})$ space (8 points) contains points which are outside the exercise boundary. Ignoring options for which warnings are issued halved the relative root-mean-square error (RMS=0.000553) and reduces the relative maximum error to 0.003394.

Table shows results similar to those presented by Ait-Sahalia & Carr. For this parameter set the lookup method compares favourably with any of the direct calculation methods. The fact that no dividends are present is not taken into account explicitly, so that 256 instead of only 64 points are used in this calculation. Taking this into account would reduce the computation time significantly.

The current implementation is still far from optimal, and could easily be speeded up. Currently 89 calls to a Neville-interpolation routine are made during every look-up, and this routine could be in-lined and tailored to the problem. A more important optimisation would be to dynamically change the number of points used for the interpolation.

The accuracy can be increased by choosing a larger grid, which comes at the cost of higher memory usage, or by splitting the domain and using a finer grid for short maturities. For particularly difficult cases, such as when some of the interpolation points lie outside the exercise boundary, or the life-span is very short, it is always possible to fall back on a slower direct calculation.

Conclusions.

The results in this note demonstrate the feasibility of using look-up tables for pricing

American options with continuous dividends. It turns out that organising the look-up table to (i) restrict it to an acceptable size, and (ii) to get high accuracy, is harder than might have been thought. A key factor for the accuracy is to take explicitly into account that the price function changes very rapidly at the exercise boundary. Accessing the table has to be thought out carefully, and cubic polynomial interpolation is used to maintain accuracy. The American put is fundamentally quite a simple option, and we find that the look-up table is broadly comparable in speed and accuracy with the best of the methods discussed elsewhere in this volume. If we were now to envisage the pricing of an American put where the assumption of a constant interest rate were to be replaced by a Vasicek model for the interest rate, the various methods used elsewhere are going to suffer badly, whereas the look-up approach is still as good; filling the look-up table will of course take much longer, but once it is filled, the interpolation method will give the same sort of speed as we have found here (thousands of options per second). In addition to its speed, a key advantage of look-up tables is that the greeks can be estimated very cheaply.

References

[1] AitSahalia, F. & Carr, P. American options: a comparison of numerical methods. This volume.

[2] Broadie, M. & Detemple. J. Recent advances in numerical methods for pricing derivative securities. This volume.

[3] Stoer, J. Einführung in die Numerische Mathematik I, 4th ed., Springer, Berlin 1983.