Interest-Rate Option Models, by Riccardo Rebanato

Chapter 8, Lattice Methods

Section 8.1. Justification of Lattice Models

Ho and Lee (HL) and Black-Derman-Toy (BDT) pioneered the use of arbitrage-free computational lattices to value interest rate options. The methodology attained popularity because:

1) Capacity to price any received set of market bonds.
2) Formal resemblance to earlier Cox-Ross-Rubinstein binomial model.

The aims of the chapter are:

1) Show rigorous justification of intuitively appealing methodology.
2) Show that the methodology, along with the techniques of forward/backward induction, offer a very efficient computational method.

Elaborating on the first aim, the author states that the reader may already be familiar with the machinery of lattice methods, but he wants to concentrate on why the machinery works – What is its theoretical underpinning? The two main tasks are:

1) Show that the averaging/discounting procedure is equivalent to evaluating the security's price as the expectation of the discounted payoff.. (Even though, in the algorithm itself, the averaging is carried out before the discounting. The author seems pretty worried about this point.)
2) Show also that the averaging/discounting procedure carries out efficiently the discounting along each path in the tree, with no duplication of effort.

So, getting going on the procedure:

- Assume a finite set of discount bonds $\{P_i\}$, $(i=1,N)$, observed in the market.
- Assume next that security prices are driven by a single factor, which is the short rate. (Also assume that either the short rate or its log can be described by a Brownian diffusion.)
- The value of the short rate today is known from the price of $P(0,0,1*\Delta t)$:

  $$r(0,0) = -ln[\ P(0,0,1*\Delta t)\ ] / \Delta t$$

  where $P(i,j,T)$ represents the price in state j at time i for the discount bond of maturity T, and $r(i,j)$ represents the time-i, state-j value of the short rate.
- Also, assume the user will know, from historical data, or from implied volatilities, the time-dependent volatility of the short rate. There is no explicit knowledge required about the drift of the short rate.
- Then, either the short rate (HL) or its log (BDT) is allowed to move up or down, with equal probability, to one of 2 states, $r(1,1)$ or $r(1,-1)$, both of which are still to be determined.
- Now, the objective is to ensure that after a move of the state variable, then the correct expectations and variances are recovered.

Author makes the observation next that it's not enough, in constructing a computational lattice, to just ensure that all the unconditional variances, given time-0 info, are matched. You also have to match all the conditional variances. So, for $k>i$, you have to match

$$E[\ (r(k)-E(r(k)|\Im_i))^2\ |\ \Im_i\ ]$$

to market info. Do this by matching, for any state *(i,j)*, to the two adjacent states, *(i+1,j-1)* and *(i+1,j+1)*. Doing this ensures that all conditional and unconditional variances come out right.

"Construction of computational lattices entails matching of all instantaneous conditional variances. (And, implicitly, expectations.)"

Now do the following: Start making statements about the variances. Initially, get the spread at the first node, between *r(1,1)* and *r(1,-1)*.

In general, for any two equal-weight samples *A* and *B*, from an arbitrary distribution, the sample variance is given by:

$$(^1/_2) * [\ (A - {}^{(A+B)}/_2)^2 + (B - {}^{(A+B)}/_2)^2\ ]$$

$$= \quad (^1/_2) * [A^2 - (A+B)A + (^1/_4)*(A+B)^2 + B^2 - (A+B)B + (^1/_4)*(A+B)^2\ ]$$

$$= \quad (^1/_2) * [A^2 + B^2 - (A+B)(A+B) + (^1/_2)*(A+B)^2]$$

$$= \quad (^1/_2) * [A^2 + B^2 - (^1/_2)*(A+B)^2]$$

$$= \quad (^1/_2) * [A^2 + B^2 - (^1/_2)*(A^2 + 2AB + B^2)$$

$$= \quad (^1/_2) * [(^1/_2)*A^2 - AB + (^1/_2)*B^2]$$

$$= \quad (^1/_2) * [(^1/_2)*(A^2 - 2AB + B^2)]$$

$$= \quad (^1/_4) * (A-B)^2$$

which implies that the sample standard deviation is given by $(^1/_2)*|A-B|$. Now, HL uses Gaussian dynamics for the short rate, and BDT uses Gaussian dynamics for the log of the short rate, so we get

$$\text{(HL)} \quad \sigma(\Delta t)^{1/2} = (^1/_2) * [r(1,1) - r(1,-1)]$$

$$\text{(BDT)} \quad \sigma(\Delta t)^{1/2} = (^1/_2) * [\ln(r(1,1)) - \ln(r(1,-1))]$$

These give the relative separation of *r(1,1)* and *r(1,-1)*. To get absolute levels for the short rates, we must impose conditions from the conditional expectation.

Now do the following. We know that $P(1,1,2\Delta t)$ should be the discounted expectation of the two-period zero coupon bond, where the expectation is taken over the possible up- and down-states at time $2\Delta t$. In particular, we should get

$$P(1,1,2\Delta t) = E_{t=\Delta t}[\ 1*\exp(-r(1,1)*(2\Delta t-\Delta t))\ |\ r(\Delta t) = r(1,1)\ ]$$

$$= \exp(-r(1,1)*\Delta t) * E_{t=\Delta t}[\ 1\ |\ r(\Delta t) = r(1,1)\ ]$$

$$= \exp(-r(1,1)*\Delta t)$$

where the first line uses the fact that we know $P(1,1,2\Delta t)$'s value at time $2\Delta t$, specifically, it will be worth $1.

By similar reasoning, $P(1,-1,2\Delta t) = exp(-r(1,-1)\Delta t)$. Now note that if we move backward one more step in time, from time Δt to time 0, and now take the expectation of $P(0,0,2\Delta t)$ over its possible up- and down-states at time Δt, then we get

$$(\dagger) \qquad P(0,0,2\Delta t) = exp(-r(0,0)*\Delta t) * [({}^1/_2)*P(1,1,2\Delta t) + ({}^1/_2)*P(1,-1,2\Delta t)]$$

Now note that

- We know $P(0,0,2\Delta t)$ and $r(0,0)$ from the market.
- $P(1,1,2\Delta t)$ and $P(1,-1,2\Delta t)$ are given above as functions of $r(1,1)$ and $r(1,-1)$, respectively.
- $r(1,1)$ and $r(1,-1)$ have a relation between them, given by the vol reasoning used previously.
- And we have assumed knowledge of the (time-dependent) vol, either from historical data or from implied volatilities.

So, the main thing to note is that, in Equation (†), there is really only one unknown, either $r(1,1)$ or $r(1,-1)$. (If you find one, then you can solve for the other.) You can then solve, analytically for HL, or numerically for BDT.

So now it's possible to do the same thing for each new discount bond. Note that, at each time i, you have

$$(HL) \qquad r(i,j+1) = r(i,j-1) + 2\sigma(\Delta t)^{1/2}$$

$$(BDT) \qquad r(i,j+1) = r(i,j-1)exp(2\sigma(\Delta t)^{1/2})$$

Then it's possible to bootstrap your way through the entire tree.

Observation from the author: Each expectation over 2 nodes at a specific time in the tree is really taking an integral by evaluating only 2 points. The points where the integrand is evaluated are chosen in such a way to give an approximation to the real integral, as the time steps become smaller. Yet, as the number of time steps grows, the points where you have to evaluate the integrand grows only linearly, not exponentially, with the number of steps.

<u>Observation from Ken:</u> It seems that what he's really talking about here is the recombinant property of the tree, rather than the 2-point property of the expectation integral. If it were not recombinant, then the number of evaluations would grow like the exponential 2^n, it seems.

<u>Section 8.2. Implementation of Lattice Models: Backward Induction</u>

The procedure outlined in Section 8.1 allows for construction of a tree such that market discount bonds are priced exactly by construction. This can be shown to be equivalent to determining the change in drift required by Girsanov's Theorem to ensure that arbitrage is avoided. Since, by matching all the market discount bonds, you have ensured that all expectations have been taken with respect to the correct probability measure implied by the chosen numeraire.

The author makes the observation that in Section 8.1, no specific use was made of the fact that when the chosen numeraire is the money-market account, then all assets earn the riskless rate of return. He says, stay tuned for Section 8.3, where it will be necessary for making statement's about Green's functions, which lead to an efficient numerical method for tree calibration. *<u>Observation from Ken:</u> I had noticed this before, that the construction seemed to* *imply that you could calibrate to any series of bonds, not just riskless bonds. It's not obvious to me how this will fail when implementing Green's functions. Waiting on the edge of my seat ...*

Next, there is an observation about whether or not exact calibration to market bonds tells us anything about the appropriateness of the volatility we are using. Author notes that, because of the construction procedure, market bond prices would have been recovered for any choice of volatility term structure.

Makes statement that the shape of the yield curve is a function of both expectation of future rates and of future volatility. (Will be shown in Chapter 9.) So, even if the model were correct in some theoretical sense, (that is, even if the world were a one-factor world with the short rate as the factor,) then the exact matching of the bond prices could give no info about the appropriateness of the chosen vol, and hence about the computed risk-adjusted drift.

In other words, the user cannot know, simply from the correct pricing of the market discount bonds, that the "correct" expectations are being carried out. Will be discussed more in Chapter 12 on BDT, where it will be shown that info about, for example, market cap prices must be considered.

Observation from Ken: *Is he really making statements here about backing out the market expectation of the 'real' drift, I wonder? Because, you know that the risk-adjusted drift should be the riskless rate. It seems that, if you then also had the 'correct' volatility, in some sense, then you could use it, moving backwards through the Girsanov's Theorem drift transformation, to get the market expectation of the real drift of the asset. Speculation on my part.*

Moving on, if we assume our volatility is appropriate, then we may price a generic security V with known payoff $V(T)$ at time-T, by moving backward through the tree, at each step taking the risk-neutral expectation of the discounted value of the security. The specific equations come out as:

$$V(n-1,j) = E[\ V(n\Delta t)\exp(-r(n-1,j))\ |\ r((n-1)\Delta t) = r(n-1,j)\ ]$$

$$= \exp(-r(n-1,j)) * E[\ V(n\Delta t)\ |\ r((n-1)\Delta t) = r(n-1,j)\ ]$$

$$= \exp(-r(n-1,j)) * (^1/_2)[\ V(n,j+1) + V(n,j-1)\ ]$$

where we are allowed to take $exp(-r(n-1,j))$ out of the expectation, since the expectation assumes knowledge of time-$(n-1)$ info. The author makes the observation that this equation entails first discounting, and then averaging, rather than doing the opposite order: taking the expectation of the discounted value.

Observation from Ken: *The author is pretty hung up on the fact that the procedure for the tree entails first averaging, then discounting, rather than taking the expectation of the discounted value. I think it's because in a previous chapter, Ch. 6 I think, he proved a theorem saying that the value of a security today is equal to the risk-neutral expectation of the discounted payoff, and he's worried about making the material of this chapter follow very rigorously from that theorem.*

Following the equations given above, and moving backward through the tree to time-0, then one can find the value today of the security V with known payoff $V(T)$ at time-T.

This procedure is referred to as backward induction, and it can be seen as the discrete-time equivalent of the backward Kolmogorov equation, which, given a particle's probability distribution at a later time T, solves for the initial position of the particle.

The counterpart to this is forward induction which, given a particle's position today, can solve for the probability distribution at a later time T. (*Observation from Ken:* *Of course you* *must know something about the evolution of the probability distribution with time.)* This is really accomplished as part of the lattice construction, specifically when finding short rates *r(i,j)* for times $i>0$. The equations which allow for this in continuous time are known as the forward (Fokker-Plank) equations, and their implementation in the context of computational lattices is described in Section 8.3, below.

Strengths of backward induction are as follows:

It is very easy to check early-exercise optimality with this method, as in the case of American options, or compound options – options on options, for instance captions. You just check the conditional expectation you obtain at each node, as the value of the option at that node, against the payoff obtained by exercising early at that node.

Observation: Author makes statement that the implementation of an *n*-period tree is really equivalent to the evaluation of an *n*-dimensional integral, with *n* conditional integrals.

Weaknesses of backward induction:

Backward induction poorly handles the case of path-dependent options. (In Chapter 10.1, the author says, it is shown how they can be used to some extent, in suitably benign cases.) The problem stems from the fact that, while at any node in the backward induction procedure, you have information about what will happen at later times, but you have no knowledge of what happened previously. All paths leading to the present node are treated the same. For path-dependent options, Monte-Carlo methods are better suited.

Section 8.3. Implementation of Lattice Models: Forward Induction

The author begins by making the statement that naively following the procedure of the last section is very costly computationally, since at each step the entire tree must be re-traversed.

Observation from Ken: *I'm not sure I agree with him on this point. Because,*

- *Each new discount bond will allow you to find the short rates at one step prior to that bond's maturity.*
- *And, just thinking about the computational part of it, you already know the short rates two time steps prior to the bond's maturity, from a previous calibration.*
- *So it seems like only one expectation computation is required, where the expectation is over the time step from 2 periods previous, to 1 period previous to the present discount bond's maturity.*

Further thought. Maybe this isn't right, because, in going from the T-2 to the T-1 step, you wind up getting some number of equations >1, which then requires you to go backward in the tree again, and again, all the way back to the original node, till you have your 1 equation which you can solve. So possibly when he says you have to 'traverse' the tree, this is what he means.

Continuing, he says that there are no-arbitrage conditions, worked out in Ch. 6 of the book, which allow for side-stepping this very laborious procedure. In the measure associated with the rolled-up money market as the numeraire, all assets, including the discount bonds, must earn the riskless rate. To utilize this, he defines the Green's function *G(i,j,s,t)*, which is the value at time-*i* and state-*j* of a security which pays $1 at time-*t* and state-*s*. This quantity is known as an Arrow-Debreu price, and it bears an analogy from physics.

In this section we deal with the case where *G(i,j,s,t)* describes a security which pays $1 if state-*s* is reached at time-*t*, and 0 otherwise. With this security in mind, then if we consider a generic security *V(0,t)*, with payoff *v(s,t)*, then *V* can be written as:

$V(0,t) = \sum_s G(0,0,s,t)v(s,t)$

And in particular, for a discount bond $P(0,T)$, we have

$P(0,T) = \sum_s G(0,0,s,T)$

So knowledge of the Arrow-Debreu prices implies knowledge of the value of a discount bond.

In getting the aforementioned efficient computational method, then, the strategy is to show the following:

- Knowledge of the Green's function for all states at time t, that is, $G(0,0,(\cdot),t)$, determines the value of Green's function for all states at the next time, $G(0,0,(\cdot),t+\Delta t)$.
- To see how this is used here, one makes use of the fact that, with the rolled-up money market as the numeraire, all assets earn the short rate over the time interval from time-t to time-$t+\Delta t$.

The argument is as follows:

- If one knew the "initial" Green's function for time-t and state-j-1, $G(0,0,j\text{-}1,t)$,
- And simultaneously knew the time-t, state-j-1 value of \$1 at time-$t+\Delta t$ and state-j, that is, $G(j\text{-}1,t,j,t+\Delta t)$,
- Then, you would immediately have the following relation for $G(0,0,j,t+\Delta t)$:

  (‡) $G(0,0,j,t+\Delta t) = G(0,0,j\text{-}1,t) * G(j\text{-}1,t,j,t+\Delta t)$

- Now note that, $G(j\text{-}1,t,j,t+\Delta t)$ must come out as $({}^1/_2)*exp[-r(j\text{-}1,t)\Delta t)]$, for some as yet undetermined short rate r.
  - Note that here, we have a factor of $({}^1/_2)$ introduced that was not there previously, because previously, we had a discount bond paying \$1 in all states, whereas here, we have an Arrow-Debreu security paying \$1 in only one state, and zero otherwise.
  - Here is where we are using the fact that, under this measure, all assets must earn the short rate r from time-t to time-$t+\Delta t$.
- Also note that $G(0,0,j\text{-}1,t)$ is known from previous calibration steps.
- Now, a note of explanation, and I don't know if I can explain this so well. The relation (‡) given above isn't really correct. To get $G(0,0,j,t+\Delta t)$, we really have to take an expectation over all possible paths which lead from time-0, state-0 to time-$t+\Delta t$, state-j, where each term in the expectation summation has the same form as the right-hand side of (‡).
- Because this is a binomial tree, the only possible states which lead to a non-zero payoff, that is, which lead to time-$t+\Delta t$, state-j, are the two states which pass through states j-1 and states $j+1$ at time-t. So the modified, correct expression comes out as:

  (‡),modified: $G(0,0,j,t+\Delta t) = ({}^1/_2)*exp[-r(j+1,t)\Delta t]*G(0,0,j+1,t)$
  $+ ({}^1/_2)*exp[-r(j\text{-}1,t)\Delta t]*G(0,0,j\text{-}1,t)$

So: Knowledge of the Arrow-Debreu prices at time-t, along with the knowledge of the short rate r at time-t, completely determines the Arrow-Debreu prices at time-$t+\Delta t$. Recall that short rates over all nodes at any time-*t* can be found if only a single one of them is know, by relations using the variances, worked out in Section 8.1.

Observation from Ken: *I'm a little confused here. Just because this was supposed to be a faster way of constructing the lattice – that is, a faster way of being able to calibrate to the market bonds. But, he says you need the present A-D prices* and *the present value of the short rate to get the next-time-step A-D prices. But, I thought the whole point of the construction/calibration procedure was to get the short rate at all the nodes. Because, once you have the short rate at all nodes, then all discount bond prices, and all option values, are trivial to get. Now, the author says we need the short rates to get the A-D prices. So I don't see the advantage to this method.*

Author goes on to say that the computational advantages afforded by this method reduce the flop count from $O(N^3)$ to $O(N^2)$, which can actually make the difference in making this method practical in real trading situations.

Conclusion of Chapter 8: It has been shown that the lattice method for option valuation is fully justified as a numerical procedure. It has also been shown that it is an efficient procedure, coming up with the price in a reasonable number of computations. And it is a simple procedure, as shown in the section on Arrow-Debreu prices. These features will be compared with the PDE and Monte Carlo methods, in Chapters 9 and 10 of the book.