Fin 514: Financial Engineering II

Lecture 4: Analysis of Binomial tree

Dr. Martin Widdicks

UIUC

Spring, 2018

Outline

- There are certain derivative pricing problems which have closed form, analytic solutions, such as European call and put options, some barrier options, some Asian options and some bonds.
- However, a far larger class of derivatives have no such solutions, the most obvious being American call and put options.
- There are three main methods for solving these problems: binomial trees (we have already seen these), finite difference methods (numerical method for the PDE solution) and Monte Carlo methods (numerical method for the probabilistic solution).
- Each of these methods has its strengths and weaknesses and we will discuss when it is optimal to use each method. Today we will look at analysis of the binomial tree

Points to consider

When we introduce the three numerical methods, you should focus on 5 key properties:

- What is the rate of convergence? As you add more iterations in the method, how quickly do we arrive at the correct answer.
- How do they converge? Is the convergence to the correct answer monotonic and if so, can we use extrapolation techniques to speed up the convergence.
- How do they deal with more sources of uncertainty? Is the increase exponential or linear. e.g. if you have an option on two stocks do you need to double the number of calculations (linear) or square it (exponential).
- How adaptable is the technique? Is it easy to adapt our method to a range of different derivative problems, including different underlying stochastic processes (e.g. stochastic volatility) and path dependency (e.g. payoff depends upon a maximum level reached, autocall feature etc.).
- Are they forward or backward techniques? This will be most important for dealing with early exercise, e.g. do you start from maturity and value backward or start from the current stock price and move forward.

Overview of main methods

Issue	Binomial	Monte Carlo	Explicit FD	C-N FD
Convergence	Proportional to the number of time steps in the tree ($1/N$).	Proportional to the square root of number of simulations ($1/\sqrt{N}$)	Proportional to the number of time steps in the grid ($1/N$)	Proportional to the square of the number of time steps in the grid ($1/N^2$).
Monotonic?	No, but can be adapted	No, cannot be adapted	No, but can be adapted	No, but can be adapted
>1 sources of uncertainty	Exponential	Linear	Exponential	Exponential
Adaptable	Yes	Yes, very	Trickier	Trickier
Backward or forward	Backward	Forward	Backward	Backward

Convergence

- When analyzing convergence we need to consider the error from a numerical scheme, if V_{exact} is the correct option value and V_n is the value from a binomial tree with N steps then:

$$Error_N = V_N - V exact$$

- To formally define convergence, there exists a constant, κ, such that for all time steps, N,

$$\mathrm{Error}_N \leq \frac{\kappa}{N^c}$$

  where c is the order of convergence. This can also be written as

$$\mathrm{Error}_N = O\left(\frac{1}{N^c}\right)$$

- As long as $c > 0$ then then V_N will converge to V_{exact}.

Quiz

Rank these orders of convergence from the fastest to the slowest:

- $c = 0.5$
- $c = 1$
- $c = 2$

Convergence for trees

- Unfortunately there is not a simple proof to show the convergence of the binomial tree to the correct option price.
- When considering European options it is simple to look at this empirically because we have an analytic expression for V_{exact} (the Black-Scholes price). By the Central Limit Theorem we also know that the prices will eventually converge as the binomial distribution converges to the lognormal distribution.
- All empirical evidence indicates that for all basic binomial models (CRR, RB etc) $c = 1$, or that V_N converges to the Black-Scholes at a rate of $1/N$. So, in general to halve the error you must double the number of time steps.
- For a rigorous proof see Leisen and Reimer (Applied Mathematical Finance, 1996).

Empirical evidence

The below diagram (from Leisen and Reimer, 1996) shows the error from the CRR model relative to $1/N$, where the upper line denotes how the error would reduce with $1/N$ convergence and the sawtooth pattern is the actual error

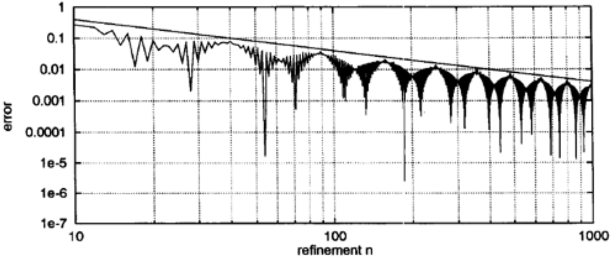


Fig. 4. Graphical representation and examination of the error bound; x-axis and y-axis with log-scale; example with CRR-model and the following selection of parameters: $S = 100$, $K = 110$, $T = 1$, $r = 0.05$, $\sigma = 0.3$, $n = 10, \ldots, 1000$.

Monotonic?

- We would like this convergence to be monotonic for two reasons:
  1. First, we would like to know that as we construct a tree with more steps we will get closer to the exact answer. This is especially important when we have no analytic value for the exact answer.
  2. Second, if the problem is computationally intensive, we can save effort by using extrapolation procedures.
- As we see from the graph on the previous slide the convergence to the exact option value looks far from monotonic for the binomial tree. We would like to investigate why this is the case.

Extrapolation

- If convergence to the exact option value is monotonic and at a known rate then there is a simple extrapolation technique.
- We have that the exact option value, V_{exact} is equal to our estimated option value from a tree with N steps, V_N, plus error terms.

  $$Vexact = V_N + \text{Error terms}$$

  We would like to make the error terms as small as possible.
- If we know the rate of convergence then we can make our errors smaller by a technique called (Richardson) extrapolation. Consider the following equations for trees with different numbers of time steps, N and M. We know that the largest error term is $1/N$ and we assume that the next error term is $1/N^2$, so we write the exact option value as

$$\begin{aligned} V_{\text{exact}} &= V_N + \frac{\kappa}{N} + \frac{\lambda}{N^2} + \text{ smaller terms} \\ V_{\text{exact}} &= V_M + \frac{\kappa}{M} + \frac{\lambda}{M^2} + \text{ smaller terms} \end{aligned}$$

Extrapolation

- To remove the first error terms we can use these simultaneous equations to determine the size of κ and thus know the first error term exactly, leaving an unknown error term of size $1/N^2$, which is a lot smaller than before. The equation becomes:

$$V_{\text{exact}} = \frac{NV_n - MV_m}{N - M} + \frac{\lambda}{(MN)} + \text{smaller terms}$$

  and so now the error depends upon approximately the number of steps squared and so is smaller than the original error.

- Next we will see what the error actually looks like for a European option:

Quiz

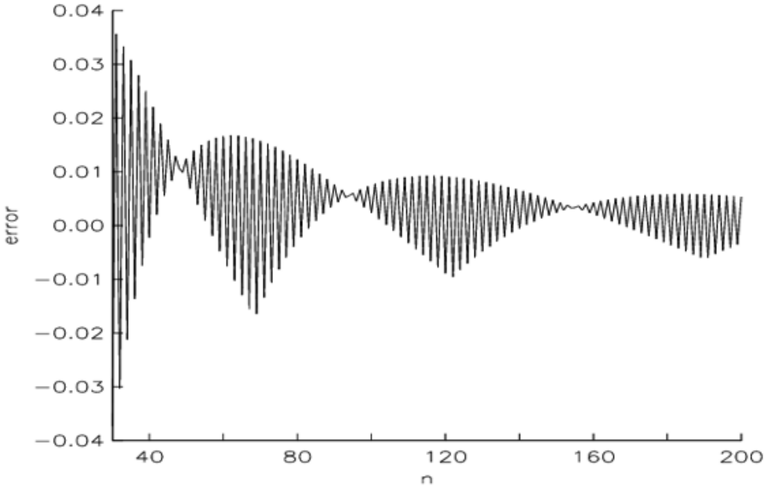
- If, as before, we assume that

$$V_{\text{exact}} = V_N + \frac{\kappa}{N} + \frac{\lambda}{N^2} + O\left(\frac{1}{N^3}\right)$$

how could you use extrapolation to make the error $O(1/N^3)$?

Error for European options

For a European option, when we increase N and plot $Error_N$ against N we see the following shape:

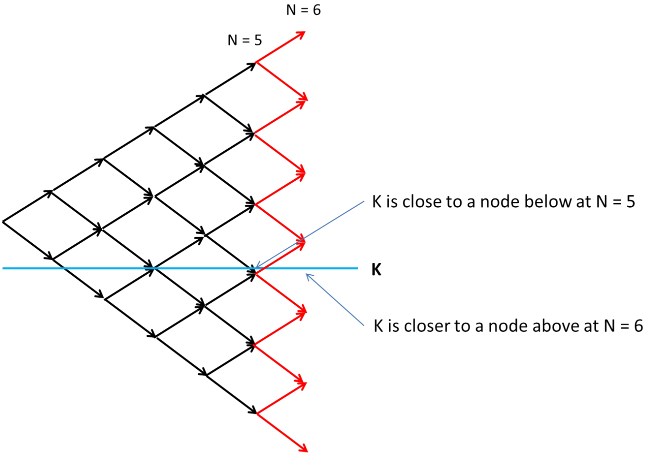


What do we see?

- We see two distinct features, the first is a **sawtoothing** and the second is **periodic humps**.
- The sawtoothing is known as the 'odd-even effect' (Omberg, Advances in Futures and Options research, 1987) where as you move from say 25 steps to 26 steps the change in V_N is very large. This happens as the final nodes (at $j = N$) in the tree move relative to the exercise price of the option, where there is a discontinuity in the option price as you move from $2N$ to $2N + 1$.
- The following diagrams explain the odd-even effect:

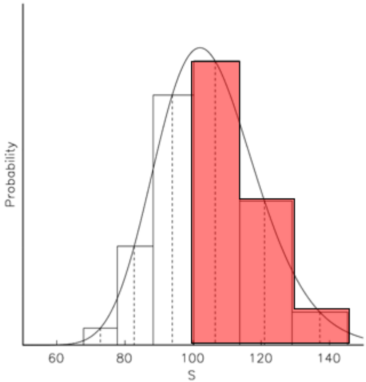
Position of K after 5 and 6 steps (slightly simplified!)

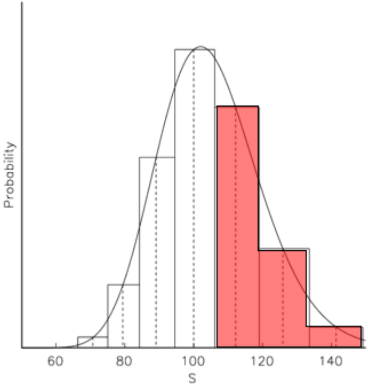
What happens to the position of K relative to the tree as we move from 5 to 6 steps. Of course as N increases the time to maturity stays fixed so the size of the steps in the six step tree should be smaller...)



Explanation of odd-even effect in detail

The binomial approximation to the normal is depicted here for trees with 5 and 6 steps. The red shading denotes which nodes contribute value to the option price if $K = 100$.





Explanation of odd-even effect

- Due to the discontinuity in the slope of the option payoff, the location of the final nodes are very important in determining V_N.
- In the first diagram, the node at 110 contributes an option value of 10 with a large probability and so this tree overvalues the European option and so V_{error} is positive . However, in the second diagram the node at 100, contributes nothing to the option value and so this tree undervalues the option and so V_{error} is negative , and we would observe the sawtoothing.

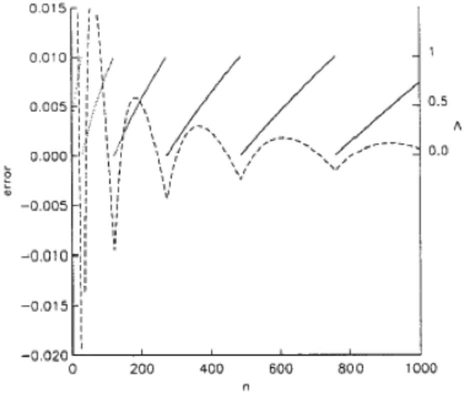
Explanation of periodicity

- The periodic humps are also a result of this as (unless $S = K$ in the CRR model) the position of the nodes relative to the exercise price change as you increase the the number of time steps N.
- If we introduce a measure Λ denoted by

$$\Lambda = \frac{S_k - K}{S_k - S_{k-1}}$$

  where S_k is the closest node above the exercise price and S_{k-1} is the node below the exercise price.
- The next diagram plots Λ against the error from the binomial tree (with only even steps to remove the odd-even effect)
- The dashed lines here denote the error and the solid lines the corresponding value of Λ. Only even numbers of steps were considered.

Explanation of periodic ringing



Quiz

If we use the CRR model $(ud = 1)$ and $S_0 = K$ what do you think the graph of N against error would look like?

- The same as above.
- One smooth, monotonically converging line.
- Alternating between two smooth monotonically converging lines.
- Something else?

Quiz solution

Types of error

- Figlewski (Journal of Financial Economics, 1999) introduced a definition to distinguish between the two types of error that one observes when pricing derivatives using binomial trees.
- First there is **'distribution error'** which arises from the binomial distribution only providing an approximation to the lognormal distribution. This is the error that converges at $1/n$, this can typically be reduced by extrapolation techniques.
- Second, and more importantly there is **'non-linearity error'** . This arises from not having the nodes in the tree or grid aligned correctly with the features for the option. For example, the strike price in a vanilla European. This can cause serious errors for more exotic options, especially those with barrier or autocall features (see problem set 3).

Removing non-linearity error

- In the case of European options the most elegant way of overcoming non-linearity error is Leisen and Reimer (1996), they use the degree of freedom in selecting q, u and d so that the tree is centered around the exercise price K, to ensure that the non-linearity error is removed (or remains constant) when N **is odd**.
- The choices which do this are as follows where N is the total number of time steps.

$$
\begin{aligned}
q &= h(d_2) \\
u &= e^{(r-\delta)\Delta t} q^* / q \\
d &= \frac{e^{(r-\delta)\Delta t} - qu}{1-q}
\end{aligned}
$$

where

Removing non-linearity error

- $$d_{1,2} = \frac{\ln(S/K) + (r - \delta \pm \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{(T-t)}}$$
  $$h(x) = \frac{1}{2} + sign(x)\sqrt{\left[\frac{1}{4} - \frac{1}{4}\exp\left\{-\left(\frac{x}{N+\frac{1}{3}}\right)^2\left(N+\frac{1}{6}\right)\right\}\right]}$$
  $$q* = h(d_1)$$

  where $sign(x)$ is the sign of the argument in h (d_1 or d_2). That is $sign(d_2)$ is a negative sign if d_2 is negative and positive if d_2 is positive.

Removing non-linearity error

- The Leisen and Reimer tree works best for **odd numbers of steps**. For extrapolation with the LR tree only perform extrapolation with N (N odd) and $2N+1$ (e.g. 51 and 103, this should work very well. Extrapolation with the even steps will also work pretty well but these have pretty larger errors over all, especially compared to odd numbers of steps.
- This method, although seemingly complex is very simple to program and provides very accurate option values. The convergence is smooth and so is amenable to standard extrapolation techniques (see problem set 3).

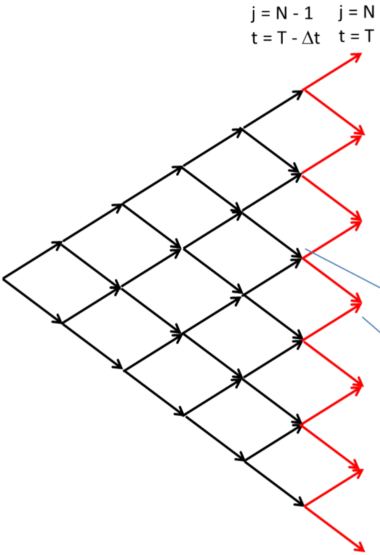
What about Americans

- Unfortunately, the Leisen and Reimer tree was designed specifically to price European options, for which we already have analytic solutions, the real test will come when evaluating American options.
- The issue of nonlinearity error is more complex in American options as we have to worry about more than simply the discontinuous payoff. At every time step there is also the early exercise boundary (which separates exercise from non-exercise).
- We do not know where this boundary will be a priori and so naturally the binomial tree will not be constructed to remove the nonlinearity error from the early exercise boundary (see problem set 3).
- There are many approaches to improving the standard CRR method for valuing American options, two of the simplest are detailed here.

Pricing American options

1. The Leisen and Reimer approach for European options also works well for American options as the largest nonlinearity error (from the discontinuous payoff) has been entirely removed. Thus, this method still provides accurate American option values and is simple to program.
2. An alternative, simple, method for pricing American options is provided by Broadie and Detemple (Review of Financial Studies, 1996) who avoid the problem of the discontinuous payoff by using a combination of the Black-Scholes formula for a European option and the CRR binomial tree.The idea is that between the penultimate timestep and expiry the continuation value of the American option is a European option with time to expiry Δt. So you can calculate the American option values at $T - \Delta t$ precisely without having any nonlinearity error from the discontinuous payoff.

BD method



Use Black Scholes formula to determine option value at $t = T - \Delta t$ with time to maturity Δt for S values $S_{N-1,i}$ for i = 0, N-1.

$$V_{N-1,i} = Ke^{-r\Delta t}N(-d_2) - S_{N-1,i}N(-d_1)$$

$$V_{N,i} = \max(K - S_{N,i}, 0)$$

Quiz

- Why does Broadie and Detemple's approach reduce nonlinearity error?

Broadie and Detemple (1996)

- If we have an N step tree with u, d and q as in CRR. Let j count time steps, so $0 \leq j \leq N$ and i count up jumps, so for a fixed j, $0 \leq i \leq j$. The Broadie and Detemple method suggests the following algorithm for pricing an American put option with N time steps and time to expiry T:

$$
\begin{aligned}
S_{j,i} &= S_0 u^i d^{j-i} \\
V_{N,i} &= \max(K - S_{N,i}, 0) \\
V_{N-1,i} &= \max(V_{h,N-1,i}, V_{x,N-1,i}) \\
V_{h,N-1,i} &= BS(S_{N-1,i}, \Delta t) \\
V_{x,N-1,i} &= K - S_{N-1,i} \\
V_{j,i} &= \max(V_{h,j,i}, V_{x,j,i}) \text{ for } j \leq N-1 \\
V_{h,j,i} &= e^{-r\Delta t}(qV_{j+1,i+1} + (1-q)V_{j+1,i}) \text{ for } j < N-1 \\
V_{x,j,i} &= K - S_{j,i} \text{ for } j < N-1
\end{aligned}
$$

Broadie and Detemple (1996)

- where

$$
\begin{aligned}
BS(S,\tau) &= Ke^{-r\tau}N(-d_2) - Se^{-\delta\tau}N(-d_1) \\
d_{1,2} &= \frac{\ln(S/K) + (r - \delta \pm \frac{1}{2}\sigma^2)\tau}{\sigma\sqrt{\tau}}
\end{aligned}
$$

- You will have a chance to analyze the accuracy of this method in PS3, and to compare it to the Leisen and Reimer method.

Exotic options

- The issue of nonlinearity error can become more pronounced for options with exotic features.
- A particular example is barrier options (see problem set 3). Typically as more and more sources of non-linearity error are introduced it becomes increasingly difficult to adapt the binomial tree to provide monotonic convergence.
- Often this is not problematic as due to increasing computing power one can just use a tree with enough time steps to overcome the problem (such as with American options).
- However, if the problem has multiple stochastic variables (such as stochastic volatility) or an interest rate derivative with a sophisticated term structure model then nonlinearity error can be a real problem.

More than one underlying asset

- There are many derivative pricing problems that require modeling more than one stochastic variable. These could be problems where we consider stochastic volatility or when the payoff from the derivative is a function of two or more underlying assets.
- There are exchange options, basket options, best of options, chooser options and a whole host of exotic derivatives which require such modeling, for example: Term Sheet from 2015.
- Here we focus on one tree approach to valuing such options when there are two underlying assets. This approach can be generalized to any number of assets (see Kamrad and Ritchken, Management Science, 1991 amongst others).

Boyle, Evnine and Gibbs

- The model presented here is from Boyle, Evnine and Gibbs (Review of Financial Studies, 1989). They assume that we have two assets S_1 and S_2, both of which are lognormally distributed with volatilities σ_1 and σ_2 respectively and the correlation between their returns over a period of time Δt is $\rho \Delta t$.
- To discretize this problem they consider a tree where at each time step the underlying asset prices (S_1, S_2) can both move up or down, giving four possible states as below:

Table 1
The continuous return distribution approximated by the four-jump process

Nature of jumps	Probability	Asset prices
Up, up	$p_1 = p_{uu}$	$S_1u_1,\ S_2u_2$
Up, down	$p_2 = p_{ud}$	$S_1u_1,\ S_2d_2$
Down, up	$p_3 = p_{du}$	$S_1d_1,\ S_2u_2$
Down, down	$p_4 = p_{dd}$	$S_1d_1,\ S_2d_2$

Boyle, Evnine and Gibbs

- In a similar way as for one underlying asset, they ensure that the first two moments of the distributions match with a (risk-neutra) joint lognormal distribution and use a CRR analogy which states:

$$\begin{aligned} u_i d_i &= 1 \\ u_i &= e^{\sigma_i \sqrt{\Delta t}} \end{aligned}$$

- This gives rise to the following probabilities:

$$p_1 = \frac{1}{4}\left(1 + \rho + \sqrt{\Delta t}\left(\frac{r - \frac{1}{2}\sigma_1^2}{\sigma_1} + \frac{r - \frac{1}{2}\sigma_2^2}{\sigma_2}\right)\right)$$

$$p_2 = \frac{1}{4}\left(1 - \rho + \sqrt{\Delta t}\left(\frac{r - \frac{1}{2}\sigma_1^2}{\sigma_1} - \frac{r - \frac{1}{2}\sigma_2^2}{\sigma_2}\right)\right)$$

$$p_3 = \frac{1}{4}\left(1 - \rho + \sqrt{\Delta t}\left(-\frac{r - \frac{1}{2}\sigma_1^2}{\sigma_1} + \frac{r - \frac{1}{2}\sigma_2^2}{\sigma_2}\right)\right)$$

$$p_4 = \frac{1}{4}\left(1 + \rho + \sqrt{\Delta t}\left(-\frac{r - \frac{1}{2}\sigma_1^2}{\sigma_1} - \frac{r - \frac{1}{2}\sigma_2^2}{\sigma_2}\right)\right)$$

Computational effort

- In the binomial tree for one underlying asset at each time step j there are $j+1$ nodes giving $(N+1)(N+2)/2$ total calculations in an N-step tree.
- As we move to the two underlying model then each step has $(N+1)^2$ nodes giving $(N+1)^2(N+2)^2/4$ total calculations, which is the square of the effort in the one underlying case.
- As you introduce k underlying assets the the total number of calculations grows exponentially to $(N+1)^k(N+2)^k/2^k$ which can become a very large number.
- Typically due to memory constraints it is difficult to get reasonable accuracy with many more then 5 underlying assets or sources of uncertainty.

‘Work ’

- So as to compare the strengths of different numerical methods Broadie and Detemple (Management Science, 2004) introduce the idea of representing the convergence as a function of work which is the computational effort required.
- Thus for a tree with N time steps and k underlying assets the work w is approximately N^{k+1} and the convergence is at the rate of $1/N$ and so the convergence can be seen as $O(w^{-1/k+1})$
- We will see this measure again when we consider Monte-Carlo methods ($O(w^{-1/2})$) and finite-difference methods (which can give $O(w^{-2/k+1})$).

Quiz

If you had a tree method which required n^k calculations for k underlying assets but converged with order $c = 2$ (like the finite difference methods) for up to how many underlyings would it be more efficient than the Monte-Carlo method?

- $d < 2$
- $d < 4$
- $d < 6$
- $d < 8$

Curtailed range

- For most options (especially American options) in more than one underlying asset a simple way of reducing the computational effort is simply to ignore the vast majority of tree calculations.
- In their curtailed range method Andricopoulos et al., (Journal of Derivatives, 2004) showed that for options on just one underlying with 1000 steps, the time saving was 87%, for options on three underlying assets with 100 steps the time saving was 91%.
- The idea is that in large trees many of the calculations are superfluous as they represent scenarios where the underlying asset price has moved in excess of ten standard deviations and so contribute nothing to the value of the option. Plus in these ranges the option will either never be exercised, or always be exercised and so we know its value.
- There are papers on COMPASS about this if anyone ever needs to implement tree calculations very fast - e.g. for high frequency trading.

Overview

- We have analyzed the binomial pricing model in detail, in general it converges at the rate of $1/N$ where N is the number of time-steps in the tree.
- However, this convergence is often monotonic due to nonlinearity error caused by discontinuities in the option price. This can be illustrated by considering the discontinuous payoff from a European call or put option.
- There are methods of overcoming this, and it is particularly important for American options where there is no analytic solution.
- Finally, we analyzed how to construct a tree for more than one underlying asset and how this effects the computational effort or work.