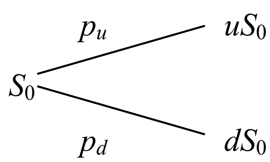
7. A Discrete-Time Model for Pricing Options

Binomial Model

An instrument whose value depends upon the value of another underlying instrument at some future time(s) is referred to by the more or less interchangeable terms "derivative," "option," or "contingent claim" (with "option" generally reserved for instruments whose payoff is always nonnegative). Many of the instruments we have dealt with so far arguably fall into this category. However, we have for the most part confined ourselves to instruments whose risks (under suitable market assumptions) can be hedged away at pricing time using a static portfolio. For some instruments, pricing and hedging require more specific information about the dynamics of the underlying's price.

Suppose, for example, that we wish to price a derivative on an equity that pays no dividends. The equity's value today is S_0, and the option's payoff will be decided at a future time T, called its expiry; at that time, the equity will have known value S_T, while the option will have known value V_T. Our desire is to determine the fair present value V_0 of the derivative.

In order to do this, we posit a very simple model of the equity's behavior: At time T, the equity will have one of two possible values: Either the equity will have moved up by some proportion u, or it will have moved down by some proportion d. The up transition will occur with probability p_u, while the down transition will occur with probability $p_d = 1 - p_u$. We can represent this model of the underlying equity in the following way:



If the option is such that only the terminal value S_T need be known in order to determine the payoff, then we can associate the payoff V_u with the up transition and V_d with the down transition. If finally we suppose that the discount factor P_T applies to cash flows occurring at T, and that this quantity is nonrandom, then we can calculate the expected present value of the option payoff…

$$V_0 = P_T\left(p_u V_u + p_d V_d\right)$$

…as the discounted expected value of the option payoff under our simple model.

Risk-neutral transition probabilities

In order to use even this simple model, we must specify its key parameters. For all models, the first step is to calibrate it: that is, to ensure that the prices it returns agree with known instrument values.

As it turns out, there is a very simple payoff whose value with which our model should certainly be made to agree: a forward contract on the equity itself. We know that the fair strike for a forward contract under the assumptions given above must be S_0 / P_T. Recall that this is the strike that makes the value of the contract at inception zero. Since the payoff is simply the difference between S_T and this strike, we can apply the above derivative pricing formula:

$$V_0 = P_T\left(p_u V_u + p_d V_d\right)$$

$$0 = P_T\left[p_u\left(uS_0 - \frac{S_0}{P_T}\right) + p_d\left(dS_0 - \frac{S_0}{P_T}\right)\right]$$

Recalling that $pu + pd = 1$, we rearrange to obtain the relationship:

$$\frac{1}{P_T} = up_u + dp_d$$

That is, our model must be constructed so that the expected return on the equity matches the risk-free return. The probabilities that satisfy this requirement are called risk-neutral transition probabilities. These are the probabilities that cause our pricing model to be arbitrage-free.

Once the transition sizes u and d are chosen, the above relationship permits us to determine the risk-neutral probabilities associated with those transitions, expressed here both in terms of the continuously compounded risk-free rate:

$$p_u = \frac{e^{rT} - d}{u - d}$$

$$p_d = \frac{u - e^{rT}}{u - d}$$

Extension of the model to multiple periods

Clearly, the single-period model above is an extremely crude approximation of an equity, whose value at any future time can take on, for our purposes, a continuum of values. A natural way of handling this circumstance is to divide the time period T into N periods of fixed length $T / N = \Delta t$. To make the model more tractable, we will at this point choose the values u and d to be the same at all points per choice of N, leading to a tree that "recombines"—i.e., when the stock price makes several u and d transitions, the order in which they occur does not matter to the resulting price. This model forms a lattice called a binomial tree stemming from S_0 and extending to time T via many forking paths.

A model of N periods gives rise to $N + 1$ possible final values S_T. Each of these final values is of the form:

$$S_{T,i} = S_0 u^i d^{N-i}, i \in \{0,1,...,N\}$$

…where i is the number of up transitions.

We will usually choose $u = 1 / d$ so that the tree not only recombines, but so that it follows constant spot levels: That is, the result of a u transition followed by a d returns the equity to the initial spot level. In this sort of a tree, the terminal spots are of the form:

$$S_{T,i} = S_0 u^{2i-N}, i \in \{0,1,...,N\}$$

…where again i is the number of up transitions.

If we imagine the equity's price path on the lattice, we understand why this is referred to as the binomial model: At each node, we imagine tossing a (potentially unfair) coin with heads probability p_u. Each time a heads is tossed, the equity moves up in the following period; otherwise, it moves down. Thus, the i index in the previous expressions can be thought of as the number of successes in N trials. Recalling the mass function of the binomial distribution…

$$P(i) = \binom{N}{i} p_u^i p_d^{N-i}$$

…we could, if we wished, evaluate the payoff at each terminal node and compute the risk-neutral expected value directly. However, we will generally use the tree by evaluating the payoff of the derivative at each terminal spot price and then moving stepwise backward through the tree, discounting at each step and arriving ultimately at the root node and the derivative's price. We choose this method rather than a direct numerical integration partly because the computation of binomial distribution probabilities can be slow for large N, but mainly because, as we will see later, evaluation of the option value at each intermediate node provides the opportunity to price some path-dependent options as well.

Pricing European options on the lattice

To illustrate the pricing method, we will consider "vanilla" options, with which you are likely already familiar. A call option confers the right, but not the obligation, to purchase an asset at a fixed strike price K; clearly, it is not optimal to do this if the asset's price is less than K. Thus, the payoff of a call option at time T is the greater of $S_T - K$ or zero, which is usually written:

$$V_{call,T} = (S_T - K)^+$$

A put option provides the corresponding option to sell the asset at strike K, making its payoff:

$$V_{put,T} = (K - S_T)^+$$

These are sometimes traded as European options, meaning that the option can only be exercised at expiry T. Thus, the terminal spot price is the only value needed to determine the payoff of the option, and therefore its value.

To illustrate the method, we will use a two-period binomial model with the following specifications:

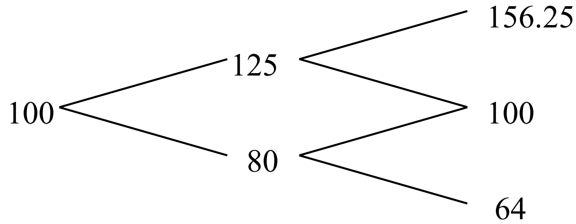
$S_0 = 100$
$u = 5 / 4$
$d = 4 / 5$
$r = 5\%$ [quoted as a rate of simple interest]
$\Delta t = 1$ year

This gives the risk-neutral transition probabilities:

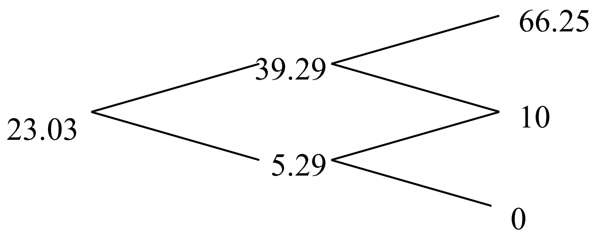
$$p_u = \frac{5}{9}$$

$$p_d = \frac{4}{9}$$

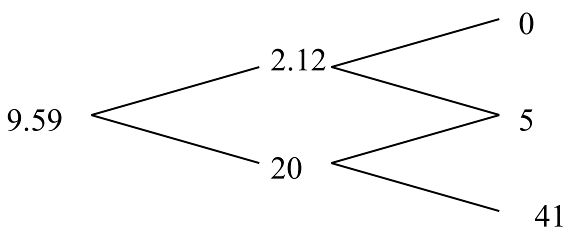
We price a European call option with strike 90 on the lattice:



We determine the terminal payoff at the right edge of the lattice, then step backwards taking discounted expectations under our risk-neutral probabilities. The result of the process is:

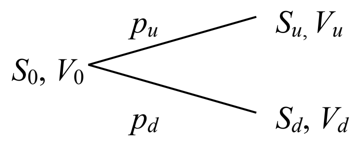


Similarly, we can price a 105-strike put on the same lattice:



Dynamic hedging on the lattice

Having a model—even a relatively simple one—of how the underlying asset moves allows us to answer other questions as well. Consider once again the simple single-period model with which we began:



To simplify matters further, we have referred to the underlying and an option V that depends upon it by the values they take on after up and down transitions.

Imagine that we start at the root node with an amount of capital X. We wish to take a position in the underlying by either buying or selling a quantity Δ; whatever capital remains will be deposited at the risk-free rate or, if we must borrow, then we will borrow at this rate as well.

Our purpose is to determine the values X and Δ that will cause our portfolio to have the same value as the option in both possible outcomes. It seems clear that we can do this, since we have two variables at hand and need only satisfy the two constraints:

$$\Delta S_u + \left(X - \Delta S_0\right)\left(1 + rT\right) = V_u$$

$$\Delta S_d + \left(X - \Delta S_0\right)\left(1 + rT\right) = V_d$$

In each case, the first term on the left-hand side is the value of the equity position after the transition, and the second term on the right-hand side is the value of the cash position growing at the rate of simple interest r. Since the cash positions are the same, it is simple to solve for Δ by subtraction:

$$\Delta S_u - \Delta S_d = V_u - V_d$$

$$\Delta = \frac{V_u - V_d}{S_u - S_d}$$

One thing that seems at least moderately interesting about this result is that the transition probabilities do not figure into it at all. More interesting is the fact that this would appear to be a numerical approximation of first derivative of the value V after the next transition with respect to S.

The value X initially looks as though it will be less enlightening. Rearranging each equation gives:

$$X = \Delta S_0 + \left(V_u - \Delta S_u\right)\left(1 + rT\right)^{-1}$$

$$X = \Delta S_0 + \left(V_d - \Delta S_d\right)\left(1 + rT\right)^{-1}$$

We then multiply through the first by p_u and the second by p_d and add:

$$X\left(p_u + p_d\right) = \Delta S_0\left(p_u + p_d\right) + \left(1 + rT\right)^{-1}\left[p_u V_u + p_d V_d - \Delta\left(p_u S_u + p_d S_d\right)\right]$$

Recalling that the two probabilities sum to 1 and rearranging a bit, we see:

$$X = \Delta S_0 - \Delta\left(1 + rT\right)^{-1}\left(p_u S_u + p_d S_d\right) + \left(1 + rT\right)^{-1}\left(p_u V_u + p_d V_d\right)$$

The second term of this is delta times the discounted risk-neutral expectation of the value of the equity after the next transition; it was a requirement of our risk-neutral transition probabilities that this value be the same as the spot price, S_0…

$$X = \Delta S_0 - \Delta S_0 + (1 + rT)^{-1}(p_u V_u + p_d V_d)$$

…and the remaining term is the discounted risk-neutral expectation of the option payoff! That is, the amount of cash required to completely replicate the option for one transition is the initial value of the option:

$$X = (1 + rT)^{-1}(p_u V_u + p_d V_d) = V_0$$

This is sometimes referred to as self-financing replication. Under the model, if we are initially given an amount of cash equal to the price of the option, then at each step we can (using the model) determine what position Δ we ought to take in the underlying to replicate the option's performance. If we rebalance this portfolio at each node, we are guaranteed that we will receive the option's payoff, and none of this requires an infusion of cash beyond the initial premium of the option.

This replication has other implications as well: In particular, it means that (again, accepting this model's view of the universe) derivatives that can be priced using this method *are not actually new assets*. Their characteristics can be completely reproduced by varying proportions of cash and the underlying equity. Any price other than the one dictated by the model would lead, in theory, to arbitrage.

In practice, all of the above statements break down to one degree or another; however, in their essentials they are followed by traders everywhere as they hedge their books. Since knowing delta shows us how to replicate the option using only cash and the stock, we can use dynamic hedging to at least reduce the P&L's associated with moves in the underlying. Also, the fact that all options on an underlying can be hedged in the same "currency"—that is, the underlying itself—means that a range of options with different strike prices and expiries can be hedged in aggregate using a single net position.

Pricing American options on the lattice

An American option is one whose exercise can take place not only at expiry, but also at any time before expiry. Thus, pricing of these options requires modeling the optimal choice, at each time throughout the contract, of whether to exercise the option or continue holding it.

This is the case where reverse induction along the tree becomes particularly useful. At each node, you calculate the value of continuation in the normal way, by taking the discounted risk-neutral expectation from the up and down nodes. With this value in hand, it becomes clear which course of action is optimal, since it can be easily compared to the intrinsic value of the option. Whichever is greater becomes the value of the option at that node, and pricing continues as usual.

To illustrate this, we choose a model with the following variables…

S0	30

Δt	0.125
r	0.02
q	0.03

…where *r* and *q* are both expressed with continuous compounding. We also choose up and down step sizes…

u	1.111895278
d	0.899365273

When the underlying pays a dividend, we must once again make sure that the risk-neutral probabilities we choose price the forward contract correctly. It can be shown using the same logic above that…

$$p_u = \frac{e^{(r-q)\Delta t} - d}{u-d}$$

$$p_d = \frac{u - e^{(r-q)\Delta t}}{u-d}$$

…making the transition probabilities and stepwise discount factor in this case…

pu	0.467630479
pd	0.532369521
step df	0.997503122

We will use this binomial model to price a call option on the underlying with expiry three months and strike 28. To do this, we price at each node, comparing the value of continuation to the value of exercise and taking the greater of the two values:

Pricing	call w/ strike 28			S	37.0893333
				V	9.089333295
		S	33.35685835	continue	0
		V	5.356858348	exercise	9.089333295
S	30	continue	5.30191695	S	30
V	2.994196601	exercise	5.356858348	V	2
continue	2.994196601	S	26.98095818	continue	0
exercise	2	V	0.932925726	exercise	2
		continue	0.932925726	S	24.2657368
		exercise	0	V	0
				continue	0
				exercise	0

Note that, at the node representing an initial up move, the optimal choice is to exercise rather than continue.

The Cox-Ross-Rubinstein parameterization

This is the set of parameters most commonly used to generate a binomial tree. We begin with the market variables:

*S*0: initial spot price

r: risk-free rate (continuous)

q: dividend rate (continuous)
T: time to expiry (years)
s: volatility

If we wish our model to consist of N time steps, then we define our up and down time steps…

$$\Delta t = \frac{T}{N}$$

$$u = e^{\sigma\sqrt{\Delta t}}$$

$$d = \frac{1}{u} = e^{-\sigma\sqrt{\Delta t}}$$

…and calculate our probabilities of up and down transitions as described above.

The main advantage of this choice stems from the fact that, in order to improve the prices returned by the model, we would prefer to be able to increase the number of time steps; however, we must do so in a way that converges to a single distribution of the asset price at T as the number of time steps increases. While this is by no means the only parameterization that satisfies the requirement, it can be shown with some persistence that it does.

It is possible, for some choices of N, that this model may contain arbitrage. Since the single-step forward price returned by the model will of necessity lie between S_u and S_d at the next time step, we must require the forward price implied by our market data to do the same:

$$e^{-\sigma\sqrt{\Delta t}} < e^{(r-q)\Delta t} < e^{\sigma\sqrt{\Delta t}}$$

If this inequality does not hold, then it is clear that one of p_u or p_d will turn out negative, while the other will be greater than 1. Such a model is not guaranteed to be stable. Fortunately, the remedy for this situation is simple: By using a large enough N, the forward can be brought to lie between the up and down points, and pricing can proceed, albeit at greater computational expense.

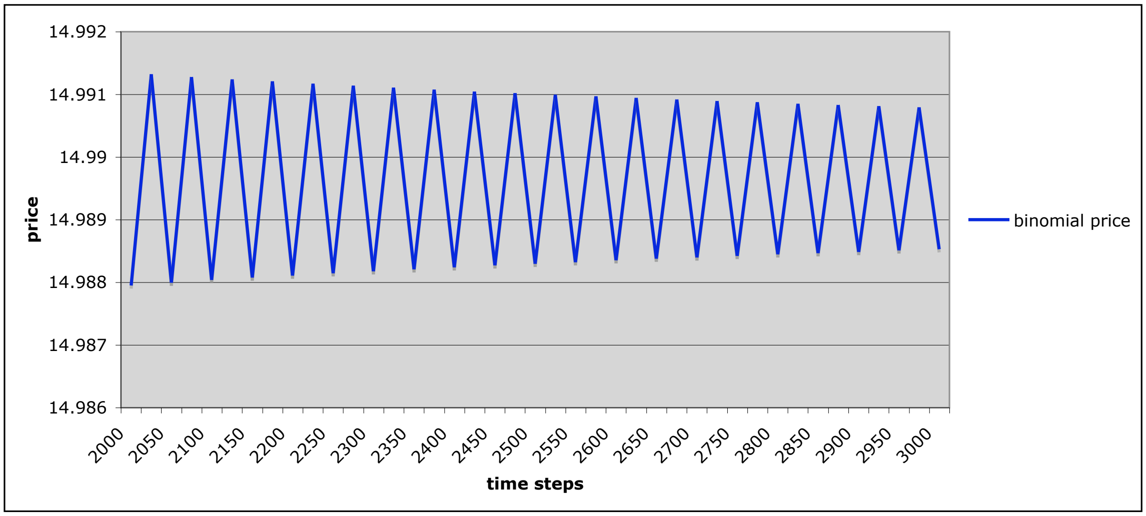
Convergence of a binomial model

It may be surprising to learn that the method detailed here, simple as it seems, is quite efficient at pricing options under the assumption of constant volatility. In using it, however, one must be aware of some of the model's oddities.

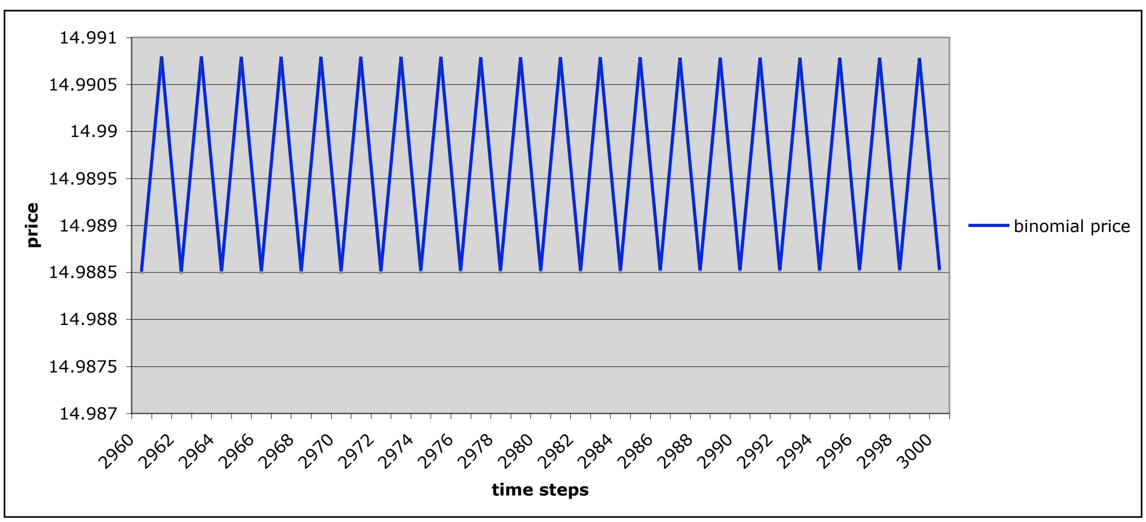
For the purposes of this illustration, we choose a European call with the following determining conditions:

$S_0 = 100$
$T = 1$ year
$\sigma = 0.35$
$r = 0.025$ (continuous compounding)
$q = 0$
$K = 100$

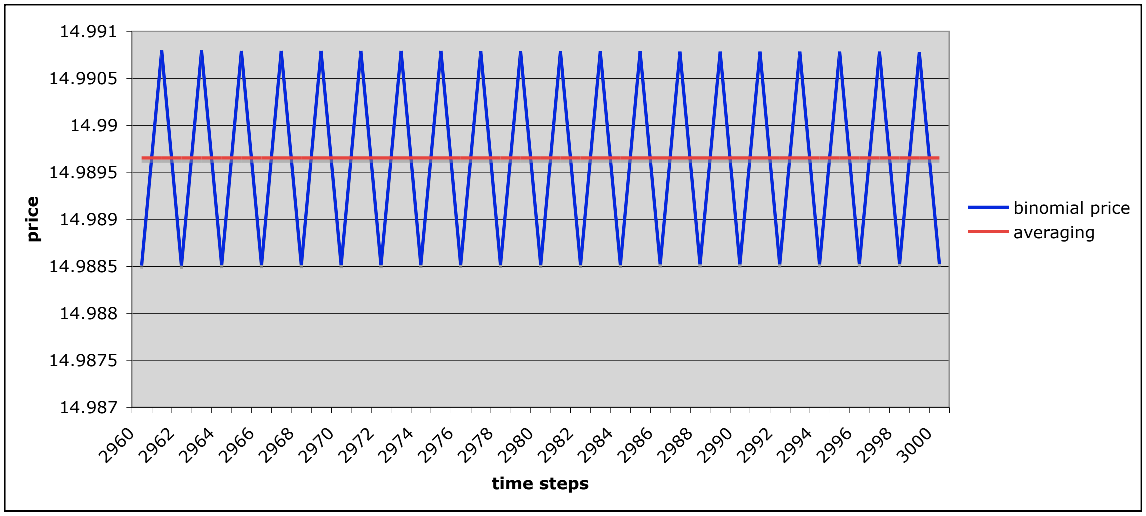
Generally speaking, as we increase the number of time steps to price this at-the-money (ATM) option, the price returned by the model converges to a single value. But the convergence is not monotonic. Here, for example, is the option above priced with N = 2000 to 3000 time steps, in increments of 25, showing the path of prices produced:



At this number of time steps, all of the values produced are quite close together—within a range of 0.003 or so—but the interesting feature is that the price produced oscillates around the value to which it evidently converges. Even numbers of time steps underprice the option, while odd numbers evidently overprice it. This phenomenon can be seen even when we consider time steps in increments of 1:

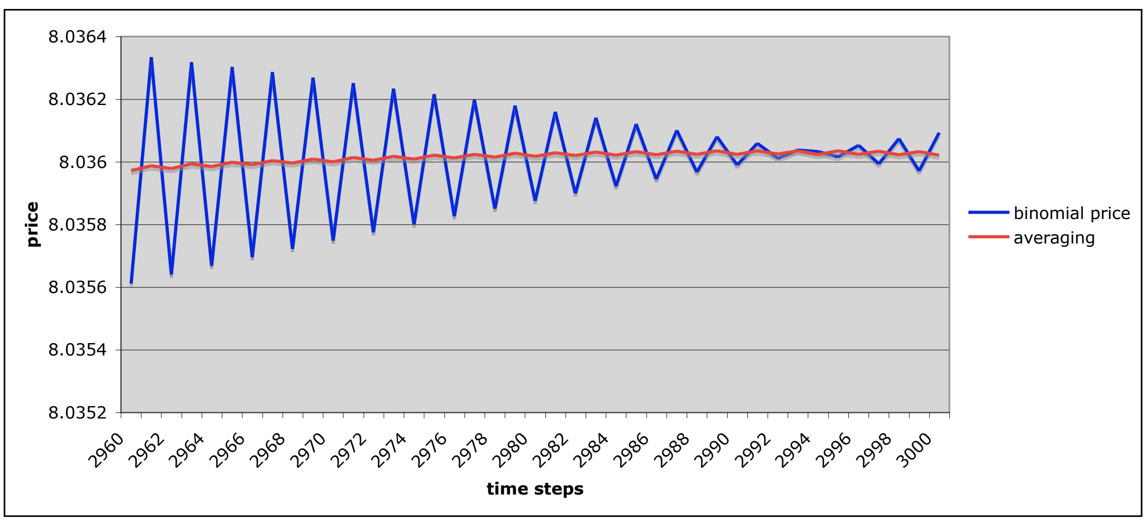


This oscillatory convergence suggests a fairly simple way to increase the quality of the estimate dramatically: Instead of pricing only for N time steps, we price also for $N + 1$ and average the two results. Certainly in the case of this ATM option the result is quite satisfactory:



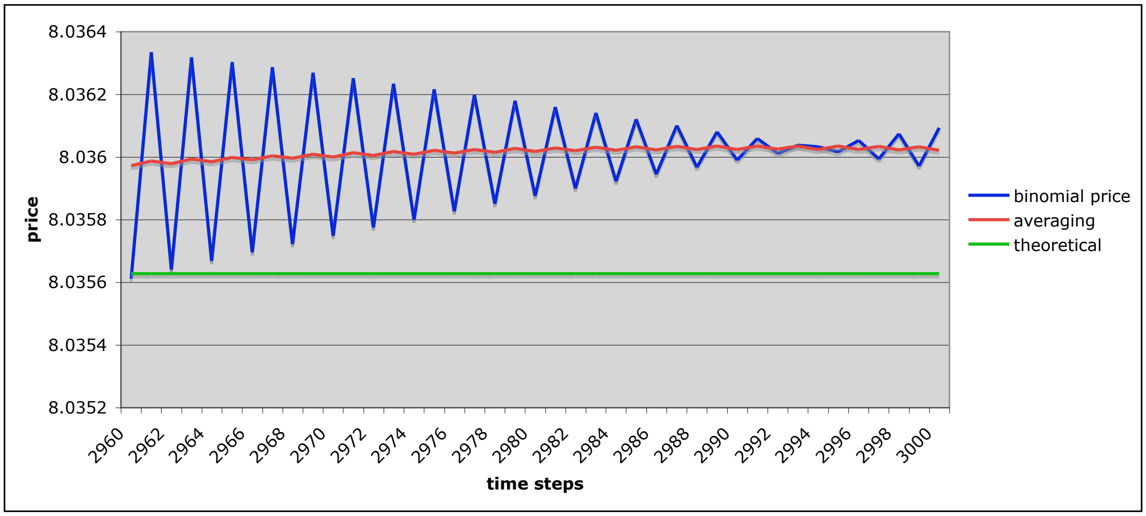
Not only does averaging greatly reduce the oscillatory behavior, it would seem to allow us to produce much more accurate results from a binomial tree with many fewer time steps than would be required otherwise.

Often in option pricing we are particularly concerned with correct pricing of out-of-the-money (OTM) options, as these are typically the most liquid ones available. If we set the strike to 120, a familiar picture emerges:

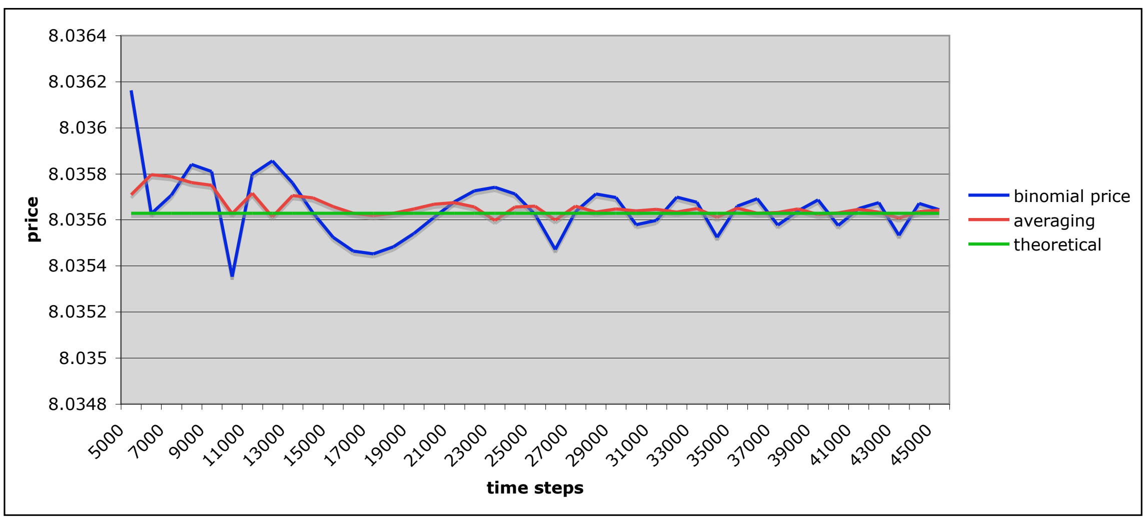


In this case, however, we see that the error does not decrease monotonically with the number of time steps. Also, unlike the equivalent ATM case, the price estimate here, even using the averaging method, seems to drift a bit as we increase the number of steps.

For this particular option, a simple pricing formula is available to determine the value to which this process ought to converge. We represent this as the theoretical price in the graph below:



Disturbingly, our iteration seems to be converging to another value entirely! We may investigate this behavior by dramatically increasing the number of time steps:



Here we see the OTM convergence behavior from 5000 to 45000 time steps, spaced 1000 iterations apart. As we can see, the convergence to a wrong value we seem to have seen above turned out to be a transitory behavior, which is fortunate. In the OTM case, convergence is still oscillatory, and the averaging method still does produce better values, but increasing the number of time steps does not improve the estimate as dramatically or reliably as one might hope, given the increasing computational expense involved.

All of these errors arise from the fact that the set of points representing the payoff time—the discretization of S_T—assigns of necessity a set of discontinuous values to the option payoff. In particular, the nodes found around the strike price have a great amount of influence on the price ultimately obtained. As N is increased, the discretization error changes; in the ATM case, this change is quite predictable, since we have constructed the tree in such a way that our initial spot price is one of the levels in the tree at all times. In the OTM case, however, the pattern of errors takes on a less regular character. Worse, for an OTM call, the discretization is sparser than it is ATM (we take an evenly spaced discretization in the log of S_T, meaning that for $K > S_0$ the spacing in S is larger than it is for $K < S_0$), leading to both worse estimates and less predictable convergence.

Richardson extrapolation
One strategy that can at times be helpful for improving upon numerical estimates is a technique called Richardson extrapolation. In general, this is a method for taking successive approximations of a quantity and, under certain assumptions about the trend of errors in the estimate, extrapolating to determine the endpoint of the successive estimation process. We will not attempt to develop the general case here, but will show how the technique works and what help it may (or may not) provide in the particular case under consideration here.

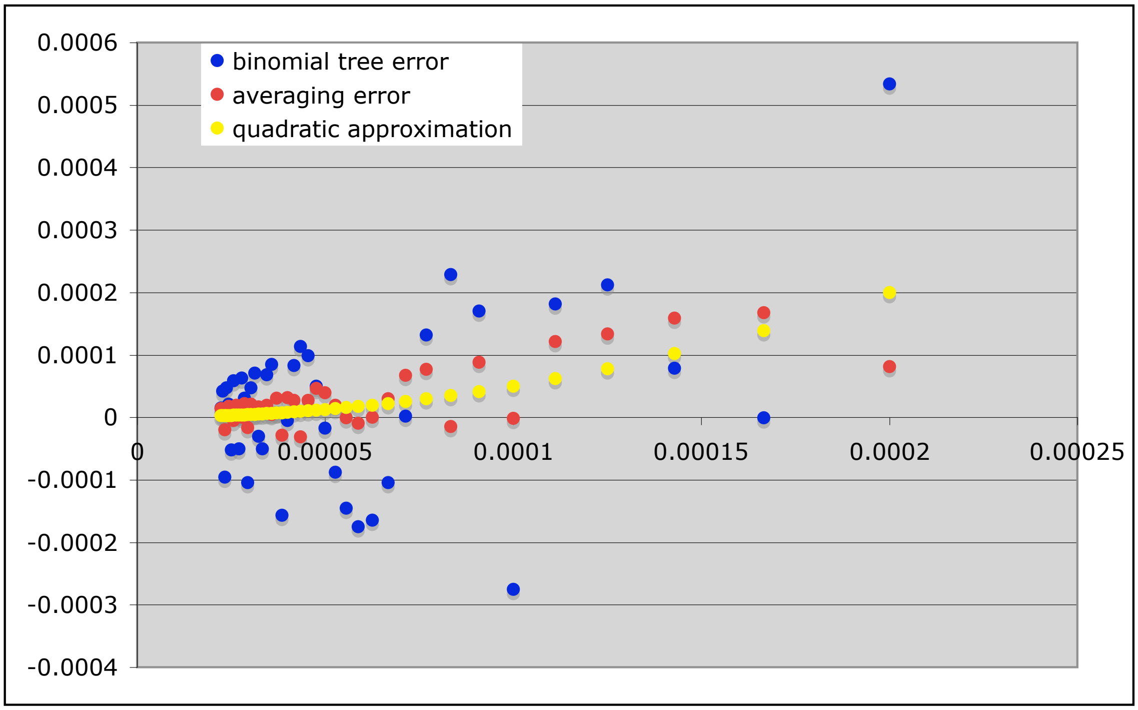
Define $A(x)$ to be the value returned by our numerical procedure for an option price, where $x = 1 / N$. Then, assuming the procedure converges to the theoretical value, what we would like most of all is to determine $A(0)$, the limit of our process at infinite N, which of course is also the theoretical value.

To do this, we make some assumption about the function governing the error in our estimate. A common choice is to treat it as approximately quadratic, meaning that our method's result is assumed to be of the form…

$$A(x) \approx A(0) + kx^2$$

…for some unknown constant k.

How good is this model of the errors for the iteration in our case? Below, we show a scatter plot of the binomial and average binomial models with $x = 1 / N$; in yellow is an arbitrary choice of k that by eye looks to be a reasonable approximation of k:



If for the moment we accept this approximate model of the errors, then consider that for some choice of x:

$$A(x) = A(0) + kx^2$$

$$A\left(\frac{1}{2}x\right) = A(0) + k\left(\frac{1}{2}x\right)^2 = A(0) + \frac{1}{4}kx^2$$

This corresponds to carrying out our numerical procedure at both N and $2N$ time steps. Then it follows that:

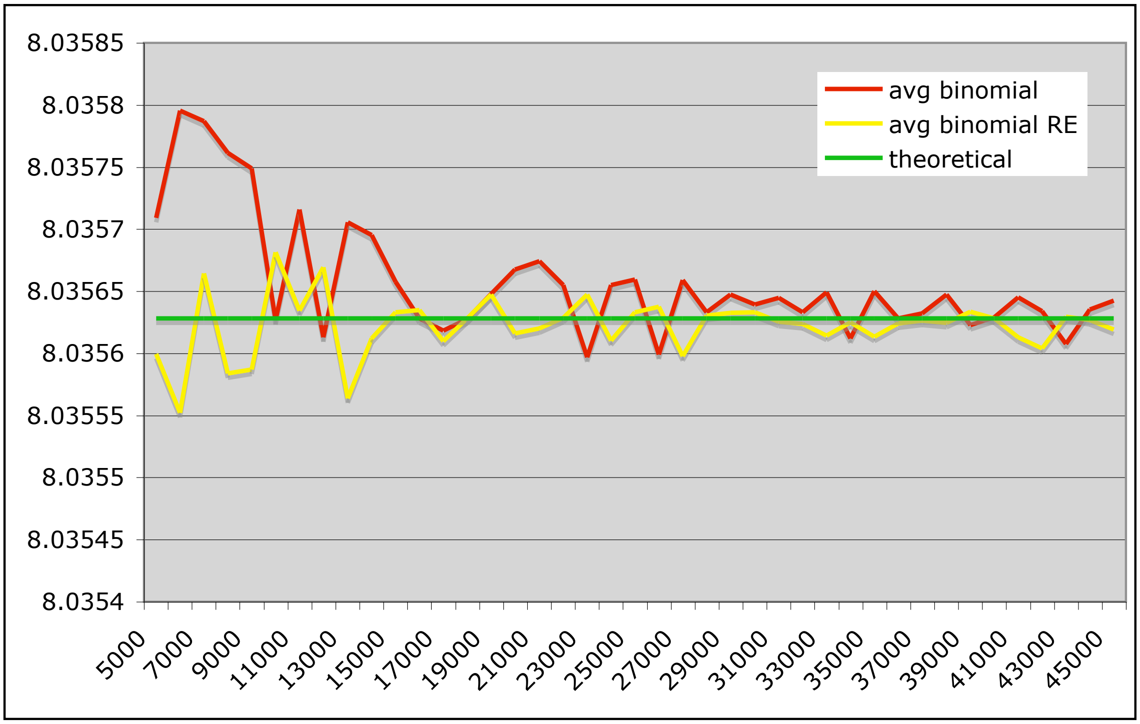
$$4A\left(\frac{1}{2}x\right) - A(x) = 4A(0) + kx^2 - A(0) - kx^2 = 3A(0)$$

So we can use these two points to estimate what value the procedure would return if it were possible to use an infinite number of time steps:

$$A_{est}(0) = \frac{1}{3}\left(4A\left(\frac{1}{2}x\right) - A(x)\right)$$

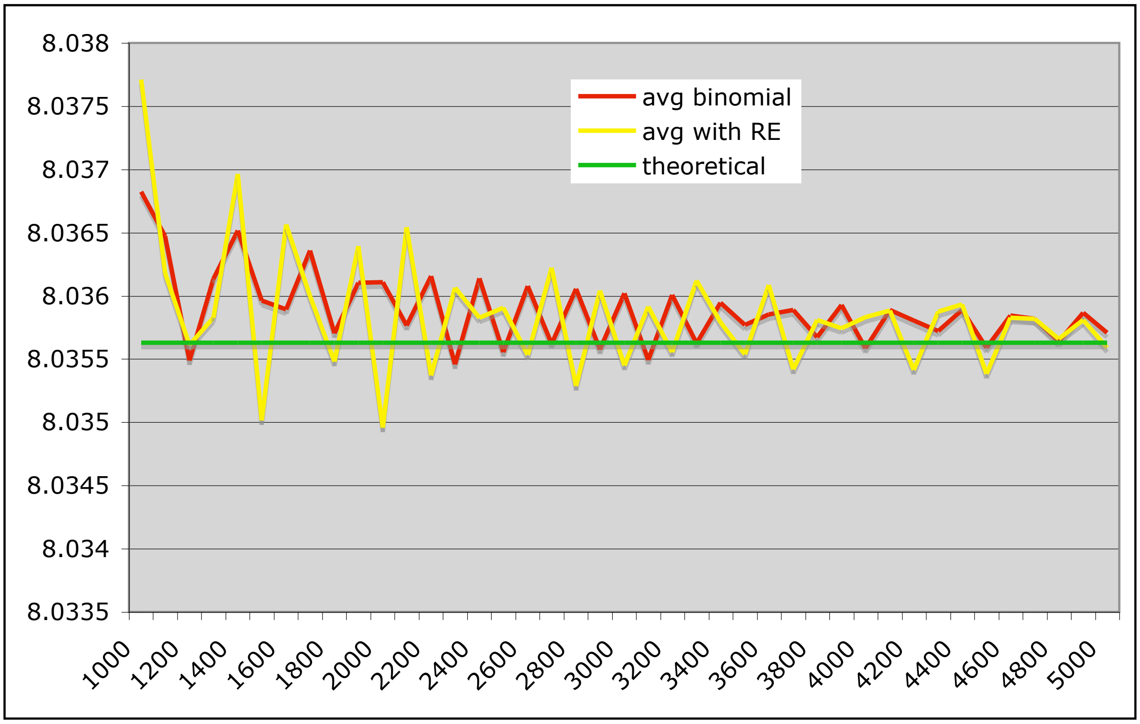
This approach does not require us to estimate the particular value of k; the quality of the estimate will depend upon how tightly our error term hews to the quadratic estimate.

In this case, it seems likely that using the estimate obtained via averaging will be better suited to Richardson extrapolation, although the oscillatory convergence of both methods leaves ample room for doubt. In the below, we compare the binomial averaging at N time steps alone (that is, the average of the result for N and $N + 1$ time steps) to the Richardson extrapolation of the averaging result for $x = 1 / N$ (that is, Richardson extrapolation using the average of N and $N + 1$ to give $A(x)$, and $2N$ and $2N + 1$ to give $A(1/2\ x)$, which obviously requires a good deal more computation):



At a glance, it seems that Richardson extrapolation can help at relatively low numbers of time steps, but at large numbers the technique adds relatively little, certainly given the computational expense of the additional pricings.

In an effort to show the tradeoffs involved using a good comparison, we propose the following test: For the binomial averaging method with N time steps, it isn't difficult to show that the number of nodes that must be processed is $(N + 2)^2$, or roughly N^2. This makes Richardson extrapolation with averaging binomial, which involves pricing at N and $2N$ time steps, of expense roughly $5N^2$. Below, we show the results of an averaging binomial pricing at N, and a Richardson extrapolation of averaging binomial at roughly 1 over the square root of 5 (0.4472) as many steps, for N in the range 1000 – 5000 for the OTM call option:



Unfortunately, it seems that once the number of computations involved is set roughly equal for the two methods, there is little evident acceleration in the convergence to the theoretical price. The oscillations in the estimates are simply too large for our quadratic approximation of the error to be useful.

Financial engineering and the choice of an appropriate method
While in this case the introduction of Richardson extrapolation was not helpful, nevertheless it serves as a good illustration of the practical realities of applying numerical methods and proposed improvements to those methods. The suitability of such an improvement can be gauged, first, by understanding the assumptions of the proposed improvement and assessing to what degree the problem at hand satisfies those assumptions. In this case, there was ample reason from the outset to be skeptical about the usefulness of Richardson extrapolation for this particular purpose.

To investigate the matter further, a suitable basis of comparison must be chosen. In this case, in order to make an apples-to-apples comparison it was necessary to find a way of equating the computational effort across the candidate methods and compare on that basis. When that was done, what at first seemed a possible advantage of the new method essentially vanished.

This is not to say that Richardson extrapolation is a worthless technique in general; far from it. In cases where convergence to the theoretical value is slow but monotonic, it can be quite helpful. But, as we have seen here, not all techniques are well suited to all problems, and a good financial engineer considers, to the extent humanly possible, the pros and cons of any given technique for a particular purpose. In this case, fortunately, there are other numerical methods available that do not suffer from the binomial tree's odd convergence behaviors and are more flexible besides. We will discuss at least one of these in a later chapter.