Fall 2006

Implied Binomial Trees in Excel without VBA

Tom Arnold
University of Richmond, tarnold@richmond.edu

Timothy Falcon Crack

Adam Schwartz

Implied Binomial Trees in Excel without VBA

Tom Arnold, Timothy Falcon Crack and Adam Schwartz
University of Richmond, University of Otago and Washington and Lee University

We implement a Rubinstein-type (1994) implied binomial tree using an Excel spreadsheet, but without using VBA (Visual Basic Application). We demonstrate both the optimization needed to generate implied ending risk-neutral probabilities from a set of actual option prices and the backwards recursion needed to solve for the entire implied tree. By using only standard Excel spreadsheet functions, and not resorting to VBA, this complicated option pricing technique is now immediately transparent to academics, students, and practitioners alike. The intuition gained from our simple spreadsheet can be applied directly to the estimation of more complicated implied trees using more advanced software. Our spreadsheet-based implementation can be used in the classroom at the advanced undergraduate level with minimal preparation.

INTRODUCTION

We demonstrate how to build an implied binomial tree (IBT) in Excel, without having to use VBA. By not using VBA, the method is immediately transparent to academics, students, and practitioners. The intuition gained from the simple Excel spreadsheet aids tremendously in the implementation of larger Rubinstein trees and other more complicated types of implied trees in more sophisticated software packages.[1] Our spreadsheet can easily be incorporated as a demonstration or as an assignment in an advanced undergraduate classroom—assuming the students are already familiar with basic Excel functions and basic multi-step binomial trees.

An IBT is a generalization of the Cox, Ross, and Rubinstein binomial tree (CRR) for option pricing (CRR [1979]). IBT techniques, like the CRR technique, build a binomial tree to describe the evolution of the values of an underlying asset. An IBT differs from CRR because the probabilities attached to outcomes in the tree are inferred from a collection of actual option prices, rather than simply deduced from the behavior of the underlying asset. These option-implied risk-neutral probabilities (or alternatively, the closely related risk-neutral state-contingent claim prices) are then available to be used to price other options.[2]

Jackwerth (1999) reviews two inter-related strands of the literature: how to infer probability distributions from option prices, and how to build IBTs. The best known

Table 1. Properties of the Best-Known Implied Binomial Tree Models

Properties of Competing Implied Binomial Tree Models	Derman/Kani 1994	Rubinstein 1994	Jackwerth 1997
IBT Constructed backwards from ending nodes?	No	Yes	Yes
Ability to use intermediate-maturity options in IBT constructions?	Yes	No[a]	Yes
Ability to use other than European-style options in IBT construction?	No	No	Yes
Requires extrapolation and interpolation in IBT construction?	Yes	No	No
Assumes all paths leading to given node are equally likely?	No	Yes	No
Approximately lognormal distribution of ending nodal probability	No	No[b]	No

[a]Practitioners are often interested in calibrating an IBT using options of multiple maturities. The Rubinstein tree that we illustrate does not allow for this. The more general Jackwerth tree (Jackwerth [1997]) does allow this.

[b]Although the objective function in Rubinstein (1994) aims to get ending nodal probabilities as close as possible to the CRR tree's ending nodal distribution (which is approximately lognormal when there are many steps in the tree), substantial deviation from lognormality is often seen.

practical methods for implementing IBTs include Rubinstein (1994), Derman and Kani (1994),[3] and Jackwerth (1997). We compare and contrast these three in Table 1.

Rubinstein (1994) is conceptually easier to implement and more stable than Derman and Kani (1994), while only slightly more mathematically restrictive than Jackwerth (1997). Rubinstein's 1994 IBT is thus the ideal candidate for our exposition. Rubinstein's IBT can be broken into three steps. In Section 1, we review these steps: build a traditional CRR tree to provide priors; solve an optimization to infer ending node risk-neutral probabilities from option prices; and finally, build the IBT using a recursive algorithm. In Section 2, we implement the model in Excel using actual option prices. Section 3 concludes and discusses directions for future research. An appendix gives some Excel solver optimization advice.

SECTION 1: BUILDING THE IBT IN THEORY

This section reviews the method for constructing a Rubinstein (1994) IBT. Rubinstein (1994, p782) gives a quadratic program to infer the posterior ending-node risk-neutral probabilities P_j given a prior P_j' based on a CRR tree, given options prices, given the initial underlying price, and given the price of the underlying at ending nodes of the CRR tree. The implied posterior risk-neutral probabilities are obtained independently of the recursion scheme used to work backwards to construct the implied tree from these probabilities. Rubinstein's prior for the distribution of the risk-neutral probabilities P_j is the distribution P_j' associated with the ending nodes of a CRR tree which, for small step size and very many steps, is approximately lognormal. Rubinstein's objective function minimizes the sum of the squared deviations of P_j from this prior while satisfying the other constraints as best as possible. Rubinstein allows for a bid-ask spread on the stock and the options, but we are using index options (on the S&P500), so we will not use the spread on the underlying. We break our Rubinstein IBT review up into three steps as follows.

Step 1: Construct a basic CRR tree.

Assume we have m European-style call options on the same underlying and of the same maturity. Use the Black-Scholes formula to infer the implied volatility σ of the nearest-to-the-money option. Use US Treasury bid and ask quotes from bills or bonds of maturities that bracket the maturity of the options to infer a continuously-compounded riskless rate R_F per annum (so a 1% rate implies $R_F = 0.01$). Break the life of the option up into n time steps of length Δt years and build a CRR tree using

$$u = e^{\sigma\sqrt{\Delta t}} \text{ and } d = e^{-\sigma\sqrt{\Delta t}}$$

Denote the nodal ending underlying asset prices of the CRR tree by S_j for $j=0,1,,,,n$, where S_0 is the lowest price, and S_N is the highest price.

Let $$p' = \frac{r-d}{u-d}$$

be the fixed risk-neutral probability of an up move over each step of the CRR tree, where

$$r = e^{R_{F_t}(\Delta t)}$$

is the riskless compounding factor per time step. Then the ending nodal risk-neutral probabilities (our priors) are simply $P_j' = \{n!/[j!(n-j)!]\}p'^j(1-p')^{n-j}$, where "n!" denotes n-factorial,[4] and $n!/[j!(n-j)!]$ is the standard binomial coefficient. Let our options have bid, ask, and strike prices C^b_i, C^a_i, and K_i, respectively for $i=1,\ldots,m$, where n is much larger than m, and let S be the spot price of the underlying. In fact, we demonstrate only $m = n = 6$ for ease of exposition.

Step 2: Optimize to find Posterior Risk-Neutral Probabilities

Assuming we have already built the CRR tree, then we have the ingredients for Rubinstein's optimization method as shown below.[5]

$$\min_{P_j} \ \Sigma_j (P_j - P_j')^2 \text{ subject to}$$

$$\Sigma_j P_j = 1 \text{ and } P_j \geq 0 \text{ for } j = 0,\ldots,n$$

$$S_0 = (\textstyle\sum_j P_j S_j)/r^n$$

$$C_i^b \leq C_i \leq C_i^a \quad \text{where } C_i = [\textstyle\sum_j P_j \ \max(0, S_j - K_i)]/r^n \quad \text{for } i=1,\ldots,m$$

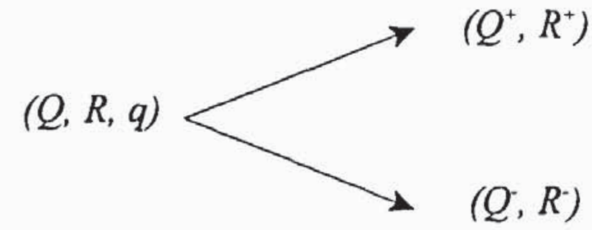
See Section 2 Step 2, for an explicit implementation using actual option prices.

Step 3: The Recursion Algorithm to Build the IBT

Rubinstein (1994, p790) gives a recursive algorithm for walking backwards through the tree using the CRR ending nodal underlying asset values and the option-implied posterior risk-neutral probabilities to build the IBT. We will have completed our IBT once we have found for each node on the tree: the path probability at that node (see below), the cumulative return at that node, and the probability of an up movement at that node (though the latter is not needed at the ending nodes). Rubinstein's recursive technique guarantees that starting from a set of positive posterior risk-neutral ending probabilities we end up with a unique (for those probabilities) arbitrage-free IBT (Rubinstein [1994, p790]).

Let us distinguish between nodal probabilities and path probabilities. Pick a node on the IBT we wish to build. If the node is in the interior of the tree, then there are multiple price paths that lead to this node. Each of these price paths has a (path) probability associated with it. The sum of these path probabilities is the total (nodal) probability that the price will arrive at this node at that time step in the tree. That is, the nodal probability is a sum of path probabilities, each of which is the probability that an individual path through the tree leads to the node. If the node is on the uppermost or lowermost branches of the tree, however, then there is exactly one price path leading to the node, and the path and nodal probabilities are identical. Both CRR and Rubinstein (1994) assume that each path leading to a node is of equal probability, and thus that the nodal probability is the path probability multiplied by the number of paths leading to that node.

Figure 1. Rubinstein's Recursion



- Recursion step zero: if the one-step ahead path probabilities Q^+ and Q^- correspond to the ending nodes of the tree, then calculate them using the appropriate $P_j/\{n!/[j!(n-j)!]\}$. Recursion step one: path probabilities are additive backwards: $Q = Q^+ + Q^-$.
- Recursion step two: path probabilities determine up probabilities $q = Q^+/Q$.
- Recursion step three: returns cumulate probabilistically: $R = [qR^+ + (1-q)R^-]/r$.

At the n^{th} time step and at the j^{th} node up from the bottom of the tree (where $j = 0$ is the bottom), there are $n!/[j!(n-j)!]$ paths through the tree leading to this node. The path probability at this node is the nodal probability divided by the number of paths: $P_j/\{n!/[j!(n-j)!]\}$.

At a particular node in the IBT, let R denote one plus the cumulative risk-neutral return through the tree to this node, so, a 20% growth in the underlying gives $R = 1.20$. Let Q denote the path probability of arriving at this node, and let q denote the probability of an up move from this node. The simple properties of path probabilities generate Rubinstein's recursion as shown in Figure 1.

Table 2 contains a summary of all the variables that appear in the paper together with the numeric values that appear in the spreadsheet in Section 2. Table 2 thus serves as a bridge between the algebra of Section 1, and the numeric implementation of Section 2.

SECTION 2: BUILDING THE IBT IN EXCEL[6]

Step 0: Collect the raw data

We choose to construct a Rubinstein IBT using European-style index options on the S&P500 (ticker SPX). The options data and SPX data in Table 3 are from a broker's web site at the close of trade April 26, 2004. The T-bill yield is inferred from the WSJ dated April 27, 2004.[7]

Step 1: Construct a basic CRR tree.

Using the 1135-strike (i.e., closest to being at-the-money) call option, the implied volatility using the Black-Scholes Model (1973) is found to be $\sigma = 0.143440$.[8] We are

Table 2. Variables Key

User-Supplied Input Variables Needed to Build the CRR Tree:

S initial spot price of the underlying [1135.53].

σ volatility (inferred from the at-the-money option) [14.3440%].

R_F continuously-compounded riskless rate (inferred from T-bills) [0.897877% per annum].

Δt length of one time step in the tree (in years) [0.024657534 (=(54/365)/6)].

n number of steps in the tree [6].

Derived Variables from the CRR Tree:

$u = e^{\sigma\sqrt{\Delta t}}$ multiplicative up growth factor [1.022779558].

$d = e^{\sigma\sqrt{\Delta t}}$ multiplicative down growth factor [0.977727793].

$r = e^{R_F(\Delta t)}$ total riskless growth rate per time step [1.000221419].

$p' = (r - d)/(u - d)$ risk-neutral up probability in CRR tree [0.499284007 at each CRR node].

Variables Derived from the CRR Tree but Needed as Inputs to the IBT:

S_j for $j = 0,1,\ldots,n$ ending nodal values of the underlying [$991.99,... $1,299.84].

P'_y for $j = 0,1,\ldots,n$ ending nodal prior (CRR) probabilities [0.015760, ..., 0.015491].

User-Supplied Input Variables Needed to Build the CRR Tree:

m number of options contracts [6].

K_i for $i = 1,\ldots,m$ strike prices of the options contracts [$1125, ..., $1160].

C^a_i for $i = 1,\ldots,m$ ask prices of the options contracts [$33.20, ..., $13.50].

C^b_i for $i = 1,\ldots m$ bid prices on the options contracts [$31.20, ..., $12.80].

Variables Derived from the IBT:

P_j for $j = 0,1,\ldots,n$ ending nodal posterior (IBT) probabilities [0.039790, ..., 0.000001].

Q the path probabilities [different for each node in the IBT; see Figure 7 Panel A].

q the implied risk-neutral up probabilities [different for each node; use Figure 1 recursion].

R the implied risk-neutral cumulative growth [different for each node; use Figure 1 recursion].

For each variable in the paper, we give the algebraic symbol and a few words of explanation. In most cases, we also give the explicit value of the variable in square brackets (as it appears in our spreadsheet example in Section 2).

Table 3. SPX Option Prices for April 26, 2004

Panel A: SPX European-Style Call Option Data at the Close of Trading			
Maturity	Days to Maturity	Strike Price	Option Bid/Ask
Sat, June 19, 2004	54 days	$1125	$31.20/33.20
Sat, June 19, 2004	54 days	$1130	$28.00/30.00
Sat, June 19, 2004	54 days	$1135	$25.00/27.00
Sat, June 19, 2004	54 days	$1140	$22.10/24.10
Sat, June 19, 2004	54 days	$1150	$17.20/18.70
Sat, June 19, 2004	54 days	$1160	$12.80/13.50
Panel B: Additional Information			
Continuous Risk-free Rate Inferred from Treasury Bills:			0.897877%
Closing Level of Standard and Poor's 500 Index:			1135.53

building a six-step tree, so we assume time step of $\Delta t = (54/365)/6 = 0.024675$ years. We then build the CRR tree using

$$S = 1135.53, \quad u = e^{\sigma\sqrt{\Delta t}} \text{ and } d = e^{-\sigma\sqrt{\Delta t}}.$$

The six-step CRR tree is shown in Figure 2; Panel A shows the values and Panel B shows the formulae.

Step 2: Optimize to find Posterior Risk-Neutral Probabilities

First, our optimization needs the "Excel Solver." To activate the solver, go to the "Tools" menu in Excel, select "Add-Ins," and select "Solver Add-in." We calculate the ending nodal risk-neutral probabilities (our priors) using the binomial probabilities $P_j' = \{n!/[j!(n-j)!]\}p'^j(1-p')^{n-j}$ entered in Cells I11:I17 of Figure 3. We populate Cells J11:J17 using simply "=I11" in Cell J11 and copying Cell J11 down Cell J17. These formulae are shown in Figure 3 Panel B, ready to be updated to become our option-implied posterior risk-neutral probabilities. Figure 3 Panel A shows the updated Column J after the optimization (Excel overwrites the formulae with numerical values).[9] Note that Figure 3 is a right-continuation of Figure 2.

Next we need to build the remaining constraints from Section 1, Step 2, above. We identify Cells J11:J19 as the variables to change when searching for the solution (see Figure 4). Then we calculate in Cell K19 the sum of squared deviations of the posterior probabilities from the priors; this is our objective function to be minimized and it is identified as the Target Cell in Figure 4. We calculate in Cell J19 the sum of the posterior probabilities to be constrained to 1.0 in Figure 4. Then we calculate in Cell S19 the implied spot price of the underlying (note that the six-period discount factor in Cell J6

Figure 2. CRR Six-Stage Binomial Tree in Excel
Panel A: Values

	A	B	C	D	E	F	G	H
1	Spot	1135.53						
2	Vol (CRR)	14.3440%						
3	Maturity	54	Days		Per Step			
4	Maturity	0.14794521	Years	u	1.02277956			
5	# Steps	6		d	0.97772779			
6	Time Step	0.02465753	Years	r	1.00022142			
7	Riskless	0.00897877	Per Annum	p'	0.49928401			
8								
9	*Step*							
10	**0**	**1**	**2**	**3**	**4**	**5**	**6**	
11	1135.53	1161.40	1187.85	1214.91	1242.59	1270.89	1299.84	**6**
12		1110.24	1135.53	1161.40	1187.85	1214.91	1242.59	**5**
13			1085.51	1110.24	1135.53	1161.40	1187.85	**4**
14				1061.34	1085.51	1110.24	1135.53	**3**
15					1037.70	1061.34	1085.51	**2**
16						1014.58	1037.70	**1**
17							991.99	**0**
18								**Node**

Panel B: Formulae

	A	B	C	D	E	F	G	H
1	Spot	1135.53						
2	Vol (CRR)	14.3440%						
3	Maturity	54	Days		Per Step			
4	Maturity	=B3/365	Years	U	=EXP(SQRT(B6)*B2)			
5	# Steps	6		D	=EXP(-SQRT(B6)*B2)			
6	Time Step	=B4/B5	Years	R	=EXP(B7*B6)			
7	Riskless	0.00897877	Per Annum	p'	=(E6-E5)/(E4-E5)			
8								
9	*Step*							
10	**0**	**1**	**2**	**3**	**4**	**5**	**6**	
11	=B1	=A11*E4	=B11*E4	=C11*E4	=D11*E4	=E11*E4	=F11*E4	**6**
12		=A11*E5	=B12*E4	=C12*E4	=D12*E4	=E12*E4	=F12*E4	**5**
13			=B12*E5	=C13*E4	=D13*E4	=E13*E4	=F13*E4	**4**
14				=C13*E5	=D14*E4	=E14*E4	=F14*E4	**3**
15					=D14*E5	=E15*E4	=F15*E4	**2**
16						=E15*E5	=F16*E4	**1**
17							=F16*E5	**0**
18								**Node**

Figure 3. Option-Implied Posterior Probabilities (Post Optimization)

Panel A: Values (Post Optimization)

	G	H	I	J	K	L	M	N	O	P	Q	R	S
6			Discount	0.9986725		Strike	1125	1130	1135	1140	1150	1160	
7						Ask	$ 33.20	$ 30.00	$ 27.00	$ 24.10	$ 18.70	$ 13.50	
8						Bid	$ 31.20	$ 28.00	$ 25.00	$ 22.10	$ 17.20	$ 12.80	
9													
10	**6**		Prior P	Posterior P	Differ^2								Spot
11	1299.84	**6**	0.015491	***0.000001***	0.000240		$ 0.0002	$ 0.0002	$ 0.0002	$ 0.0002	$ 0.0001	$ 0.0001	0.00
12	1242.59	**5**	0.093214	***0.015113***	0.006100		$ 1.7771	$ 1.7016	$ 1.6260	$ 1.5504	$ 1.3993	$ 1.2482	18.78
13	1187.85	**4**	0.233703	***0.440514***	0.042771		$27.6876	$25.4851	$23.2825	$21.0799	$16.6748	$12.2696	523.27
14	1135.53	**3**	0.312498	***0.235042***	0.005999		$ 2.4750	$ 1.2998	$ 0.1246	$ -	$ -	$ -	266.90
15	1085.51	**2**	0.235046	***0.186629***	0.002344		$ -	$ -	$ -	$ -	$ -	$ -	202.59
16	1037.70	**1**	0.094288	***0.082910***	0.000129		$ -	$ -	$ -	$ -	$ -	$ -	86.04
17	991.99	**0**	0.015760	***0.039790***	0.000577		$ -	$ -	$ -	$ -	$ -	$ -	39.47
18		**Node**											
19		Sum	1.000000	***1.000000***	***0.05816***	Model	$31.8975	$28.4488	$25.0000	$22.6005	$18.0502	$13.5000	1135.53
20						Ask-Model	***1.30***	***1.55***	***2.00***	***1.50***	***0.65***	***0.00***	***0.00***
21						Model-Bid	***0.70***	***0.45***	***0.00***	***0.50***	***0.85***	***0.70***	

Figure 3 (continued)

Panel B. Formulae (Pre Optimization)

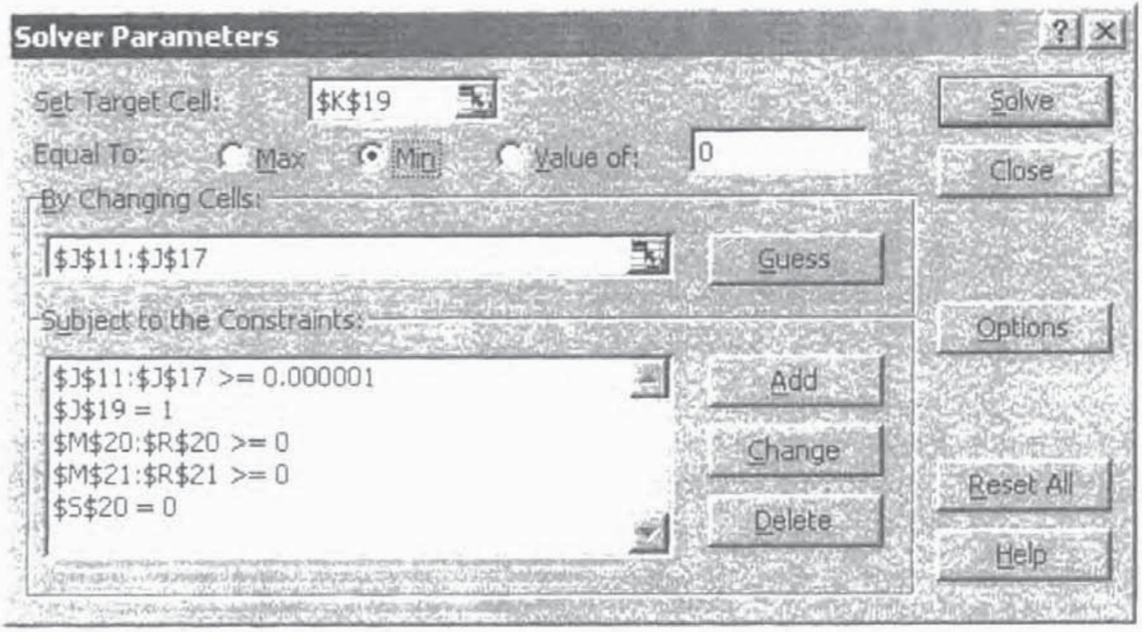
	G	H	I	J	K
6			Discount	=EXP(-B7*B4)	
7					
8					
9					
10	**6**		Prior P	Posterior P	Deviation^2
11	1299.34	**6**	=COMBIN(G10,H11)*(E7^H11)*((1-E7)^(G10-H11))	***=I11***	=(J11-I11)^2
12	1242.59	**5**	=COMBIN(G10,H12)*(E7^H12)*((1-E7)^(G10-H12))	***=I12***	=(J12-I12)^2
13	1187.85	**4**	=COMBIN(G10,H13)*(E7^H13)*((1-E7)^(G10-H13))	***=I13***	=(J13-I13)^2
14	1135.53	**3**	=COMBIN(G10,H14)*(E7^H14)*((1-E7)^(G10-H14))	***=I14***	=(J14-I14)^2
15	1085.51	**2**	=COMBIN(G10,H15)*(E7^H15)*((1-E7)^(G10-H15))	***=I15***	=(J15-I15)^2
16	1037.7	**1**	=COMBIN(G10,H16)*(E7^H16)*((1-E7)^(G10-H16))	***=I16***	=(J16-I16)^2
17	991.99	**0**	=COMBIN(G10,H17)*(E7^H17)*((1-E7)^(G10-H17))	***=I17***	=(J17-I17)^2
18		**Node**			
19		Sum	=SUM(I11:I17)	***=SUM(J11:J17)***	***=SUM(K11:K17)***
20					
21					

Note that the Excel function "=COMBIN()" when applied to two inputs *n* and *j* returns n!/[j!(n - j)!], which is the standard binomial coefficient, as mentioned in the paper.

Figure 3 Panel B: Formulae (Pre Optimization)...continued

	L	M	N	O	P	Q	R	S
6	Strike	1125	1130	1135	1140	1150	1160	
7	Ask	33.2	30	27	24.1	18.7	13.5	
8	Bid	31.2	28	25	22.1	17.2	12.8	
9								
10								Spot
11		=MAX($G11-M$6,0)*$J11	=MAX($G11-N$6,0)*$J11	=MAX($G11-O$6,0)*$J11	=MAX$G11-P$6,0)*$J11	=MAX($G11-Q$6,0)*$J11	=MAX($G11-R$6,0)*$J11	=G11*J11
12		=MAX($G12-M$6,0)*$J12	=MAX($G12-N$6,0)*$J12	=MAX($G12-O$6,0)*$J12	=MAX$G12-P$6,0)*$J12	=MAX($G12-Q$6,0)*$J12	=MAX($G12-R$6,0)*$J12	=G12*J12
13		=MAX($G13-M$6,0)*$J13	=MAX($G13-N$6,0)*$J13	=MAX($G13-O$6,0)*$J13	=MAX$G13-P$6,0)*$J13	=MAX($G13-Q$6,0)*$J13	=MAX($G13-R$6,0)*$J13	=G13*J13
14		=MAX($G14-M$6,0)*$J14	=MAX($G14-N$6,0)*$J14	=MAX($G14-O$6,0)*$J14	=MAX$G14-P$6,0)*$J14	=MAX($G14-Q$6,0)*$J14	=MAX($G14-R$6,0)*$J14	=G14*J14
15		=MAX($G15-M$6,0)*$J15	=MAX($G15-N$6,0)*$J15	=MAX($G15-O$6,0)*$J15	=MAX$G15-P$6,0)*$J15	=MAX($G15-Q$6,0)*$J15	=MAX($G15-R$6,0)*$J15	=G15*J15
16		=MAX($G16-M$6,0)*$J16	=MAX($G16-N$6,0)*$J16	=MAX($G16-O$6,0)*$J16	=MAX$G16-P$6,0)*$J16	=MAX($G16-Q$6,0)*$J16	=MAX($G16-R$6,0)*$J16	=G16*J16
17		=MAX($G17-M$6,0)*$J17	=MAX($G17-N$6,0)*$J17	=MAX($G17-O$6,0)*$J17	=MAX$G17-P$6,0)*$J17	=MAX($G17-Q$6,0)*$J17	=MAX($G17-R$6,0)*$J17	=G17*J17
18								
19	Model	=SUM(M11:M17)*J6	=SUM(N11:N17)*J6	=SUM(O11:O17)*J6	=SUM(P11:P17)*J6	=SUM(Q11:Q17)*J6	=SUM(R11:R17)*J6	=SUM(S11:S17)*J6
20	Ask-Model	***=M7-M19***	***=N7-N19***	***=O7-O19***	***=P7-P19***	***=Q7-Q19***	***=R7-R19***	***=S19-B1***
21	Model-Bid	***=M19-M8***	***=N19-N8***	***=O19-O8***	***=P19-P8***	***=Q19-Q8***	***=R19-R8***	

Figure 4. Solver Parameters



is used); Cell S20 calculates the deviation of implied spot from actual spot, and this is constrained to zero in Figure 4. We use Cells M11:R17 to calculate future option payoffs for each final node, and then we sum them and discount to give the IBT option prices in Cells M19:R19. These model prices should be a non-negative distance below the ask and above the bid; these distances are calculated in Cells M20:R21 and are constrained to be non-negative in Figure 4. Figures 3 and 4 show all the Solver Parameters and Solver Options, respectively that we use to generate the posterior probabilities shown in Cells J11:J17. Note that we constrain the posterior probabilities to be positive by bounding them below with an arbitrarily small number.

After all the constraints are entered as solver parameters (as per Figure 4), we may alter the solver options (we have not done so for this run). Then click on the Solve button. If the solver finds a solution, as in Figure 6, then we have the option to highlight and save some reports (we have highlighted "Sensitivity" in Figure 6). Clicking OK in Figure 6 updates our prior probabilities by overwriting Cells J11:J17 with the solution.

Our solution places all the IBT model option prices within the spread; two of them are binding (the 1135-strike at the bid, and the 1160-strike at the ask).[10]

It is interesting to notice how the option-implied ending node probabilities compare with the associated risk-neutral probabilities (which we use as our priors). Figure 3 Panel A reveals a relative skew toward the lower nodes in the posterior risk-neutral probabilities in Cells J11:J17 as compared to the prior probabilities in Cells I11:I17. We leave it as an exercise for the reader to demonstrate that this skew in probabilities is intimately linked to the volatility skew revealed if our IBT model option prices (in Cells M19:R19) are used to infer implied volatilities using the Black-Scholes formula and if these implied volatilities are plotted against the strikes in Cells M6:R6.

Figure 5. Solver Options

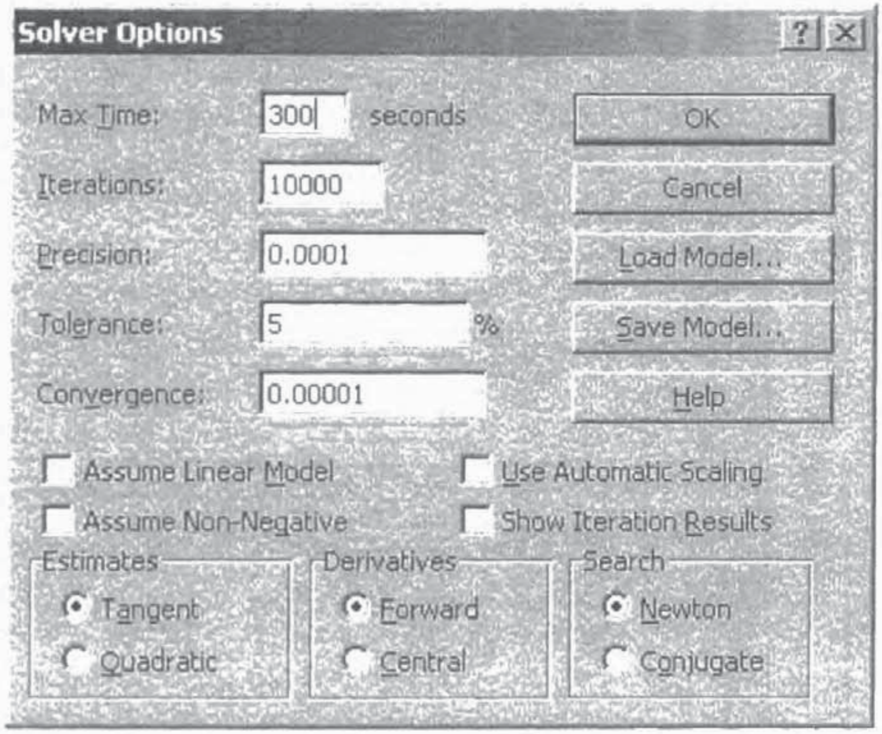
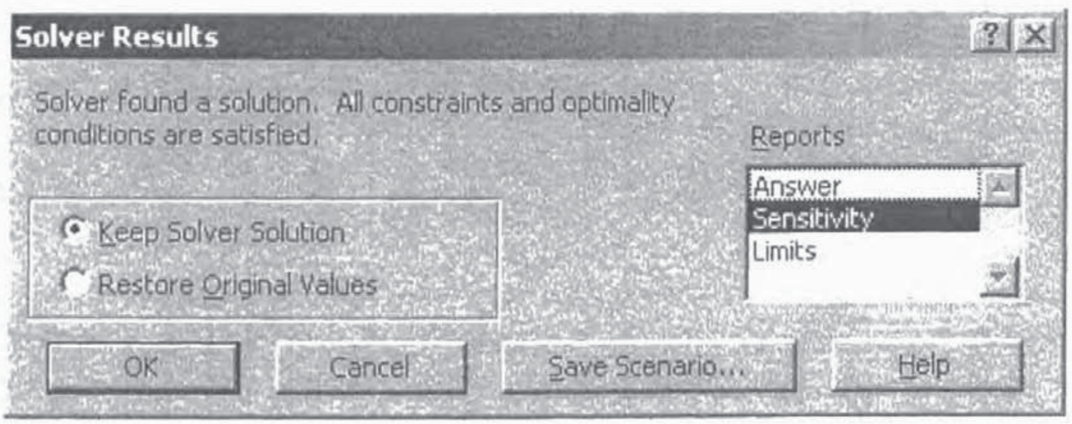


Figure 6. Solver Results



Step 3: The Recursion Algorithm to Build the IBT

We now have to build the IBT. We have the initial and ending asset values (these are the same as in the CRR tree), and the ending node probabilities. To price other options using our IBT, we need to find for each node on the tree the path probability at that node, the cumulative return at that node, and the probability of an upward price movement at that node (though the latter is not needed at the ending nodes).

Figure 7. The IBT Path Probabilities Q

Panel A: Path Probability Values

	A	B	C	D	E	F	G	H
22	*Step*							
23	**0**	**1**	**2**	**3**	**4**	**5**	**6**	
24	1.000000	0.499821	0.245733	0.107413	0.034406	0.002520	0.000001	**6**
25		0.500179	0.254088	0.138320	0.073006	0.031887	0.002519	**5**
26			0.246091	0.115768	0.065314	0.041120	0.029368	**4**
27				0.130323	0.050454	0.024194	0.011752	**3**
28					0.079869	0.026260	0.012442	**2**
29						0.053608	0.013818	**1**
30							0.039790	**0**
31								***Node***

Panel B: Path Probability Formulae

	A	B	C	D	E	F	G	H
22	*Step*							
23	**0**	**1**	**2**	**3**	**4**	**5**	**6**	
24	=B24+B25	=C24+C25	=D24+D25	=E24+E25	=F24+F25	=G24+G25	=J11/COMBIN(G23,H24)	**6**
25		=C25+C26	=D25+D26	=E25+E26	=F25+F26	=G25+G26	=J12/COMBIN(G23,H25)	**5**
26			=D26+D27	=E26+E27	=F26+F27	=G26+G27	=J13/COMBIN(G23,H26)	**4**
27				=E27+E28	=F27+F28	=G27+G28	=J14/COMBIN(G23,H27)	**3**
28					=F28+F29	=G28+G29	=J15/COMBIN(G23,H28)	**2**
29						=G29+G30	=J16/COMBIN(G23,H29)	**1**
30							=J17/COMBIN(G23,H30)	**0**
31								***Node***

We use Rubinstein's recursion as described in Section 1 Step 3, above (see our Figure 1) to build Figure 7. Note that we have shown only the path probabilities *Q*, but it easy to generate the up probabilities *q* from these, and then to derive the cumulative returns *R*. Once we have the *R* values, we can infer from them the values of the underlying throughout the IBT. It should be noted that although the value of the underlying asset in the CRR tree and the IBT coincide at both the initial node and ending nodes, the underlying asset values at intermediate nodes typically differ. Similarly, although the CRR tree assumes a constant probability of an up move p', the IBT generalizes this to allow for non-fixed up move probability *q*.

Pricing any other option (be it exotic, American-style, etc) is now as easy for the IBT as it is for the CRR tree. The implied tree for the underlying asset values (found using the initial underlying asset value and the cumulative returns *R*) determines the option's payoffs. You need only weight these payoffs using the risk-neutral nodal probabilities (found by multiplying path probabilities *Q* by binomial coefficients), and then discount at the riskless rate. If the option is American-style, then exercise decisions need to be made as per the traditional CRR tree. If the option is exotic with path-dependent payoffs, then you have the same problems as a standard CRR tree.

SECTION 3: CONCLUSION AND FUTURE RESEARCH

We show how to implement Rubinstein's 1994 Implied Binomial tree using only the standard features of an Excel spreadsheet. By not having to resort to VBA, the technique is immediately more accessible (and comprehendible) to academics, students, and practitioners. With very little preparation, this technique can be introduced into the advanced undergraduate classrooms. The intuition gained from our simple Excel spreadsheet aids tremendously in the implementation of larger Rubinstein trees and other more complicated types of implied trees in more sophisticated software packages.

There are several directions for future work. First, the objective function that uses the sum of squared differences between the CRR tree ending node risk-neutral probabilities (our prior) and the IBT option-implied probabilities (our posterior) is suggested by Rubinstein (1994, p782) and Chriss (1997), but other specifications can be imposed. Jackwerth and Rubinstein (1996, p1620) discuss several alternatives. In particular, Jackwerth and Rubinstein's "smooth" objective function adds a regularity condition (i.e., smoothness) that others have found to be very well behaved in other applications (e.g. Arnold et al [2005a, 2005b]). This can easily be implemented and illustrated in Excel. Second, Jackwerth (1997) looks at Generalized Implied Binomial trees (GIBT) as a set of nth-step probabilities and a weighting scheme for assigning those probabilities to the (n-1)st step. This gives identical results to Rubinstein (1994) if path probabilities are assumed equal. Our paper could be restated using Jackwerth's notation, and then generalized directly to his GIBT. Jackwerth's notation is in fact slightly easier than Rubinstein's. It allows generalization from a linear weight function (implicitly assumed by CRR and Rubinstein [1994] to distribute probability backwards through the tree) to a piecewise linear weight function assumed by Jackwerth (1997). Third, there are even implied trinomial trees (Derman et al [1996], Haug [1997]), and these may be amenable to Excel implementation.

Appendix: Excel Solver Advice

If Excel finds no feasible solution, try altering the precision either up or down (under "Options" on the solver menu). Allowing for option and spot bid-ask spreads rather than simply mid-spread values may improve estimation by widening the target. Note that if the precision (under "Options" on the solver menu) is coarser than the small positive lower bound on the probabilities, and if Excel's solver arrives at a negative probability, then Excel may still report that all constraints are satisfied. More time steps improve estimation. We notice, though we do not show, that moving from a six-step tree to a 20-step tree to a 100-step tree produces increasingly well-behaved volatility smiles (using IBT model prices and Black-Scholes to infer the implied volatilities). That is, when there are few time steps, the volatility smile or skew sometimes has an unrealistic kink or dip, often when the IBT model price for that particular strike is binding at the bid or ask.[11] If there is no feasible solution, then the solution returned by Excel needs to be treated with caution. It is probably best to try changing the precision of the solver first, and if that does not work, increase the number of steps in the tree, or include bid-ask spreads. We found that Excel macros prove very helpful when constructing a 100-step tree; without them, we would be copying and pasting many times which is error-prone and time consuming. As mentioned previously, we view Excel as a means to understanding the model and we think large trees should be implemented using other more sophisticated software. Finally, note that if an arbitrage opportunity exists among the prices for the options and the underlying, then the IBT may fail to converge. This can happen with closing price data where the option price and the underlying price might not be sampled synchronously.

ENDNOTES

[1] We do not recommend that practitioners use our Excel method for estimating large or complicated IBTs. Rather, we view our Excel method only as a means to understanding the model.

[2] Stephen Ross asserts that options should be spanned by state-contingent claims (Ross [1976]). One implication is that with sufficient structure, we should be able to infer state-contingent claim prices or a probability density from options prices (Rubinstein [1994, p779]).

[3] Haug (1997) gives a nice discussion of Derman and Kani (1994).

[4] For example, 4!=4*3*2*1=24, and 5!=5*4*3*2*1=120. We set 0!=1.

[5] For simplicity, we have omitted the bid-ask spread and the dividend yield on the underlying. See Rubinstein (1994, p782) for details.

[6] A copy of our spreadsheet is available at *www.KelleySchool.com/papers.html* for download.

[7] We linearly interpolate between the continuously-compounded yields on the T-bills of maturities bracketing the maturity of the option (51 and 58 days respectively). These continuous yields are calculated using the midpoints of the WSJ quoted bid-ask spreads as per Cox and Rubinstein (1985, p255).

[8] We use the option's bid-ask spread midpoint of $26.00, assume that maturity is 54/365=0.147945 years, and ignore the dividend yield. Whether you use a Black-Scholes implied standard deviation or a CRR tree implied standard deviation makes little difference when there are many steps. If using Black-Scholes implied standard deviation causes a problem, then there are not enough steps in the tree.

[9] Note that even though Figure 3 Panel A shows the post-optimization spreadsheet values, Cells I11:I17 retain their pre-optimization values as check values for the reader.

[10] We checked our numerical results using FORTRAN and the LCONF linear constrained optimization routine. The FORTRAN option prices agree with those reported here up to one thousandth of a penny (i.e., the fifth decimal place). The FORTRAN posterior risk-neutral probabilities agree with those reported here up to the fifth decimal place. For all practical purposes, the EXCEL and FORTRAN results are indistinguishable. We assume that VBA would produce results comparable to our FORTRAN results, and thus be indistinguishable from those reported here using EXCEL.

[11] More time steps improves pricing in general, but also, more time steps removes any discreteness-induced differences between implied volatilities from Black-Scholes and from an IBT.

REFERENCES

Arnold, Tom, Timothy Crack and Adam Schwartz. "Inferring Physical Probability Distributions from Option Prices" (June 22, 2005a). http://ssrn.com/abstract=749904

Arnold, Tom, Timothy Crack and Adam Schwartz. "Valuing Real Options using Implied Binomial Trees and Commodity Futures Options" forthcoming, *Journal of Futures Markets.*

Black, Fisher and Myron Scholes. "The Pricing of Options and Corporate Liabilities." *Journal of Political Economy*, vol. 81 (May/June, 1973), 637–659.

Chriss, Neil. *Black-Scholes and Beyond: Option Pricing Models.* (New York, 1997), McGraw-Hill.

Cox, John, Stephen Ross and Mark Rubinstein. "Option Pricing: A Simplified Approach." *Journal of Financial Economics,* vol. 7 no. 3 (1979), 229–263.

Cox, John C., and Mark Rubinstein. *Options Markets,* (Englewood Cliffs, NJ, 1985), Prentice-Hall.

Derman, E., and I. Kani. "The Volatility Smile and Its Implied Tree," Goldman Sachs Quantitative Strategies Reports, (January, 1994).

Derman, E., I. Kani and N. Chriss. "Implied Trinomial Tree of the Volatility Smile," *The Journal of Derivatives*, vol. 3 no. 4 (1996), 7–22.

Haug, Espen Gaarder. *The Complete Guide to Option Pricing Formulas*, (New York, NY, 1997), McGraw-Hill.

Jackwerth, Jens Carsten and Mark Rubinstein. "Recovering Probability Distributions from Option Prices." *Journal of Finance*, vol. 51 no. 5 (1996), 1611–1631.

Jackwerth, Jens Carsten. "Option-Implied Risk-Neutral Distributions and Implied Binomial Trees: A Literature Review." *Journal of Derivatives*, vol. 7 no. 2 (1999), 66–81.

Ross, Stephen. "Options and Efficiency." *Quarterly Journal of Economics*, vol. 90 no. 1 (February, 1976), 75–89.

Rubinstein, Mark. "Implied Binomial Trees." *Journal of Finance*, vol. 49 no. 3 (1994), 771–818.