Extending Binomial Pricing

Fabio Cannizzo

Extending Binomial Pricing

- So far we have learned how to price European payoffs with binomial trees, and how to construct a risk neutral tree consistent with a GBM (or ABM) for a non-dividend paying stock
- Now we extend this framework to cover for
  - more payoffs
  - more possible behaviors of the underlying asset
  - more ways to construct the tree (CRR)

Continuous Dividend

- We go back to the introduction to risk neutral pricing, but we modify the arguments to account for the fact that the stock pays dividends
- At first, let's assume the stock pays continuous dividend at a rate y (for example, this could be a foreign currency, which pays the foreign interest rate)

Continuous Dividend

- We want to price a derivative contract paying $f(S_T)$ at time T
- Like before, let's consider a simple economy where a **currency** exchange rate at time T can assume only two values: S^+ and S^-

$$
S \nearrow S^+ \rightarrow f(S^+)
$$

$$
S \searrow S^- \rightarrow f(S^-)
$$

The value of the derivative contract in the end is just the cash flow implied by its payoff

Continuous Dividend

- Let's assume we can trade the foreign currency (EUR) at rate *S*. The foreign currency pays interest at rate *y* (in EUR). We can also invest or borrow the domestic currency (USD) at fixed interest rate *r*.
- We want to construct a portfolio investing β in the domestic risk free rate and buying δ units of the foreign currency, which replicates the payoff of the derivative contract in all the possible final states of the world
- Note that, because the foreign currency pays interests, δ units of the foreign currency will become at the end of the period $\delta e^{y\Delta t}$ units, and their value will depend on the new value of the exchange rate *S*

$$\Pi = \delta S + \beta$$

Continuous Dividend

- The no arbitrage linear equations become:

$$\begin{cases} V^{+} = f\left(S^{+}\right) = \delta\, S^{+} e^{y\,\Delta t} + \beta\, e^{r\,\Delta t} \\ V^{-} = f\left(S^{-}\right) = \delta\, S^{-} e^{y\,\Delta t} + \beta\, e^{r\,\Delta t} \end{cases} \Rightarrow \begin{cases} \delta = \dfrac{V^{+} - V^{-}}{S^{+} - S^{-}} e^{-y\,\Delta t} \\ \beta = \left(V^{+} - \delta\, S^{+} e^{y\,\Delta t}\right) e^{-r\,\Delta t} \end{cases}$$

Continuous Dividend

- By no arbitrage the price of the derivative must be the price of its replicating portfolio

$$\Pi = \delta\, S + \beta$$

$$= \delta\, S + \left(V^{+} - \delta\, S^{+} e^{y\,\Delta t}\right) e^{-r\,\Delta t}$$

$$= e^{-r\,\Delta t} \left(\frac{V^{+} - V^{-}}{S^{+} - S^{-}} \left[S e^{(r-y)\Delta t} - S^{+} \right] + V^{+} \right)$$

$$= e^{-r\,\Delta t} \left[\frac{S e^{(r-y)\Delta t} - S^{-}}{S^{+} - S^{-}} V^{+} + \left(1 - \frac{S e^{(r-y)\Delta t} - S^{-}}{S^{+} - S^{-}} \right) V^{-} \right]$$

$$e^{-r\,\Delta t} \left[q V^{+} + (1-q) V^{-} \right] \quad \text{where} \quad q = \frac{S e^{(r-y)\Delta t} - S^{-}}{S^{+} - S^{-}}$$

Continuous Dividend

- If we indicate up movements with $S^+=uS$ and down movements as $S^-=dS$, we notice that, as before, the probability q does not depends neither on the payoff, nor on the position in the tree

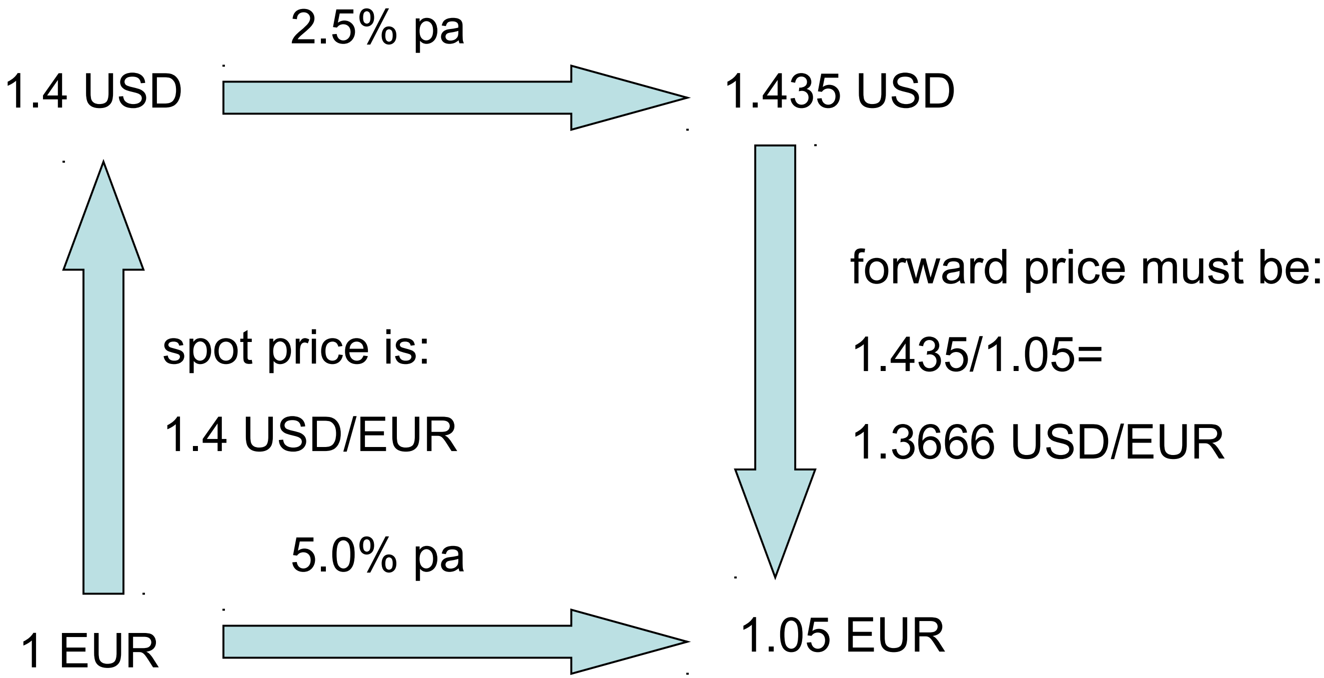
$$q=\frac{S\,e^{(r-y)\Delta t}-S^-}{S^+-S^-}=\frac{e^{(r-y)\Delta t}-d}{u-d}$$

- What is the expectation of the stock price in one period?

$$E\left[S_T\right]=\left[quS+(1-q)dS\right]=S\left[q(u-d)+d\right]=S\left[\frac{e^{(r-y)T}-d}{u-d}(u-d)+d\right]=Se^{(r-y)T}$$

- We obtained again the **forward price** (as it could be shown using a cash and carry argument) of the stock

Example of Cash And Carry



$$F = S\frac{1+r^{USD}}{1+r^{EUR}}$$

Note that in this example, for simplicity, we are using annually compounded rates instead of continuously compounded rates

Continuous Dividend

- So the previous algorithm needs only minimal changes. Namely, when we compute χ_1, we need to replace $\exp(r\Delta t)$ with $\exp((r\text{-}y)\Delta t)$, as $(r\text{-}y)$ is the **risk neutral drift** of a currency.
- The model can be used for anything which has a total drift different from the risk free rate (e.g. convenience yield, stock dividends, storage costs)
- It is assumed we **know the yield *y*** in advance!

Alternative Construction Procedures

- So far we always used the approach recommended by Cox, Ross, Rubenstein, i.e. we removed the extra degree of freedom by adding the condition $u=1/d$
- Another popular approach consist in forcing p=0.5 (Jarrow, Rudd 1982)
- While the first approach generates trees where the central node at each step is equal to the original node, the second approach will introduce a **tilt** in the tree (upward if $r-y>0$, downward if $r-y<0$)

JR Tree Calibration

- Using the same notation as before:

$$p = 0.5$$

$$pu + (1-p)d \quad = \chi_1 \qquad u = 2\chi_1 - d$$

$$pu^2 + (1-p)d^2 = \chi_2 \qquad d^2 - 2\chi_1 d + 2\chi_1^2 - \chi_2 = 0$$

$$u = \chi_1 + \sqrt{\chi_2 - \chi_1^2}$$

$$d = \chi_1 - \sqrt{\chi_2 - \chi_1^2}$$

Comparing *JR* with *CRR*

N	4
T	2
Spot	10
Strike	10
RiskFree	7%
Yield	0
Volatility	30%

d	0.801983
u	1.24691
p	0.525113

CRR

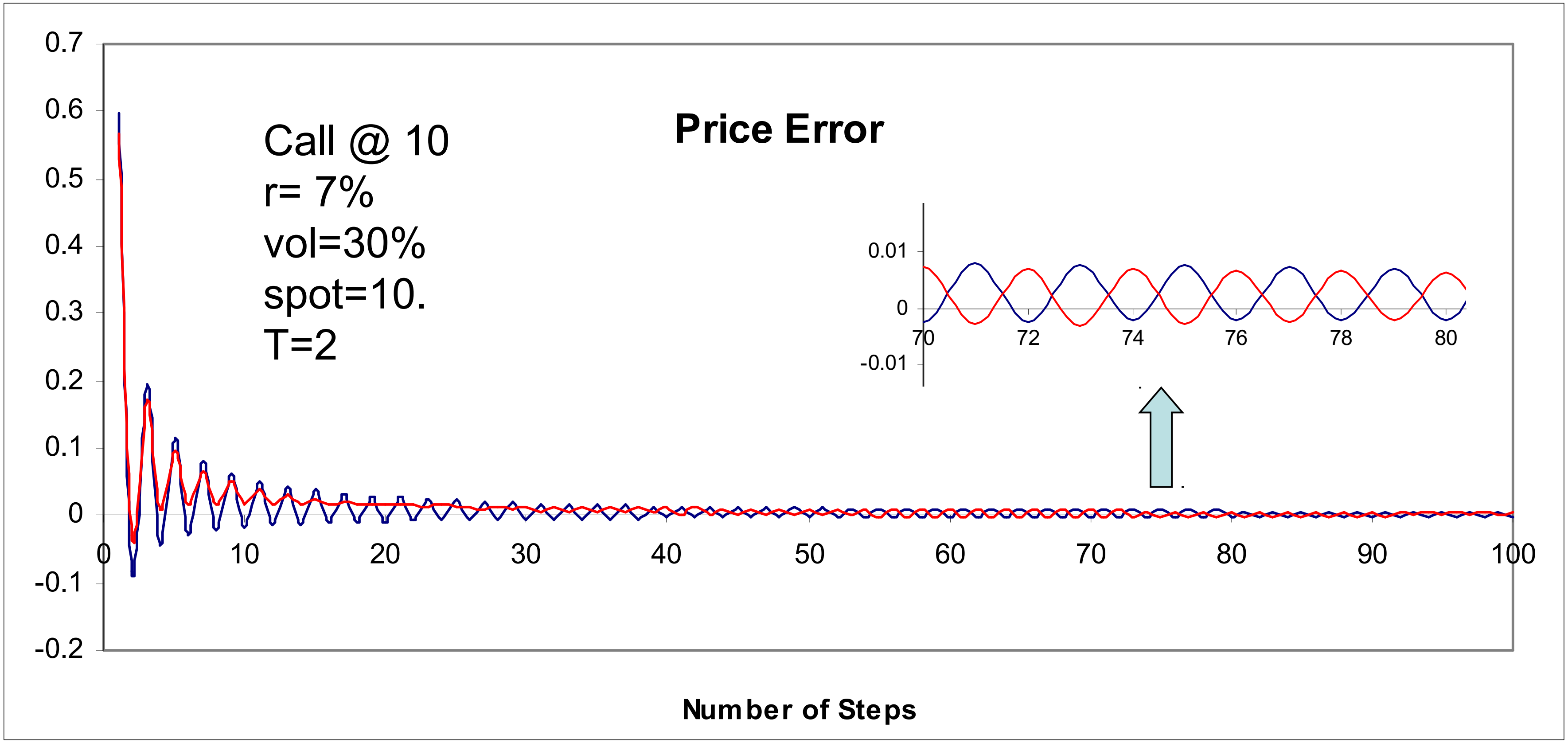
				24.17	
			19.39		
		15.55		15.55	
	12.47		12.47		
10.00		10.00		10.00	no tilt
	8.02		8.02		
		6.43		6.43	
			5.16		
				4.14	

d	0.813437
u	1.257803
p	0.5

P=0.5

				25.03	
			19.90		
		15.82		16.19	
	12.58		12.87		
10.00		10.23		10.47	tilt up
	8.13		8.32		
		6.62		6.77	
			5.38		
				4.38	

JR vs CRR Tree Example



- Similar convergence speed and pattern
  - crr_vs_jr.mu

Tilting the Tree

- A comparison of the two techniques (CRR vs JR) suggests us that it is possible to play with the geometry of the tree
- For instance we could specify a condition $ud=\alpha$ where α is chosen in such a way that some final node will match the strike of a call option or perhaps the forward price
- Such techniques have been used in literature to reduce oscillations, which is an undesired feature when using **extrapolation** techniques (e.g. Richardson)

Time Varying Yield and IR

- So far we dealt with situation where yield and discount rates and volatilities where constant
- But in practice often models calibrated to market exhibit some locality in parameters
- A generalized GBM, with time dependent IR and yield has the form

$$\frac{dS}{S} = \left[r(t) - y(t) \right] dt + \sigma\, dW$$

Time Varying Yield and IR

- Now the expected return of the stock changes at every time step
- We can compute *u*, *d* and *p* independently at each time step, matching the **local** expectation and second moment of the geometric price return for that period
- For the first period I can use u=1/d, as before, or something different. In general, let's say $u/d=\alpha$
- For successive periods we need to make sure that the tree is recombining, e.g. that $u_t d_{t-1}=u_{t-1} d_t$

Time Varying Yield and IR

- In period *t*, using the same notation as before:

$$u / d = u_{t-1} / d_{t-1} = \alpha$$

$$pu + (1-p)d \quad = \chi_1 \qquad p = \frac{\chi_1 - d}{u - d}$$

$$pu^2 + (1-p)d^2 = \chi_2 \qquad d = \frac{(\alpha+1)\chi_1 - \sqrt{(\alpha+1)^2 \chi_1^2 - 4\alpha\chi_2}}{2\alpha}$$

- Note that χ_1 and χ_2 are the local property for period *t*, obtained via bootstrapping
- α is constant throughout the tree

Time Varying Yield and IR Example

B_t is the value of a money market account, assumed deterministic

$$\frac{dB_t}{B_t} = r(t)\,dt,\ \ B_0 = 1 \qquad \text{we know the function } r(t)$$

the price of a zero coupon bond paying 1 at time T is $E_0\left[\frac{1}{B_T}\right] = \frac{1}{B_T}$

we can compute B_{t_1} and B_{t_2}

$$B_{t_1} = B_0 \exp\underbrace{\left(\int_0^{t_1} r(u)\,du\right)}_{r_{0,1}t_1}$$

$$B_{t_2} = B_0 \exp\underbrace{\left(\int_0^{t_2} r(u)\,du\right)}_{r_{0,2}\,t_2} = B_0 \exp\underbrace{\left(\int_0^{t_1} r(u)\,du + \int_{t_1}^{t_2} r(u)\,du\right)}_{r_{0,1}\,t_1 + r_{1,2}\,(t_2 - t_1)} \qquad \text{(boot-strapping)}$$

and from this we compute $r_{0,1}$ and $r_{1,2}$. Because we are interested in the properties of the GBM only at discrete steps, we can replace the arbitrary function $r(t)$ with the function stepwise constant $r(t) = \begin{cases} r_{0,1}, & 0 < t < t_1 \\ r_{1,2}, & t_1 < t < t_2 \end{cases}$

Time Varying Yield and IR Example

N	4
T	2
Spot	10

RiskFree	5%	8%	12%	15%
Yield	3%	3%	3%	3%
Volatility	30%	30%	30%	30%
χ1	1.01005	1.025315	1.046028	1.061837
χ2	1.067159	1.099659	1.144537	1.179393
d	0.806194	0.818378	0.83491	0.847528
u	1.240397	1.259143	1.284579	1.303993
p	0.469496	0.469496	0.469496	0.469496

				26.16207
			20.06304	
		15.61837		17.00399
	12.40397		13.03993	
10		10.15113		11.05171
	8.061936		8.47528	
		6.597708		7.183038
			5.508492	
				4.668602

- We could verify that the first and second cumulative moments implied by the tree at every time step match the theoretical ones
- The technique could be used also for “tiny” changes in volatility

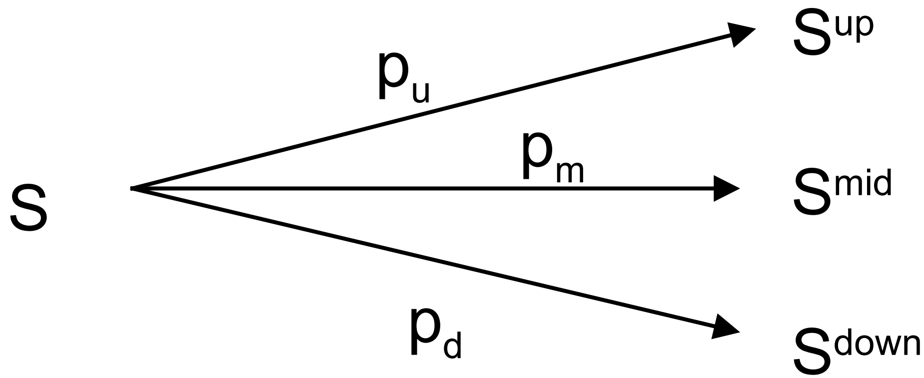
Trinomial Model

- A commonly used alternative to binomial trees are **trinomial trees**, where at every time step the stock can take three values: up, mid and low (Boyle 1986)
- The no arbitrage argument used before no longer applies, because we cannot match 3 possible states of the world with just 2 securities (3 equations in 2 unknowns). I.e. it is not a **complete market** model

Trinomial Model

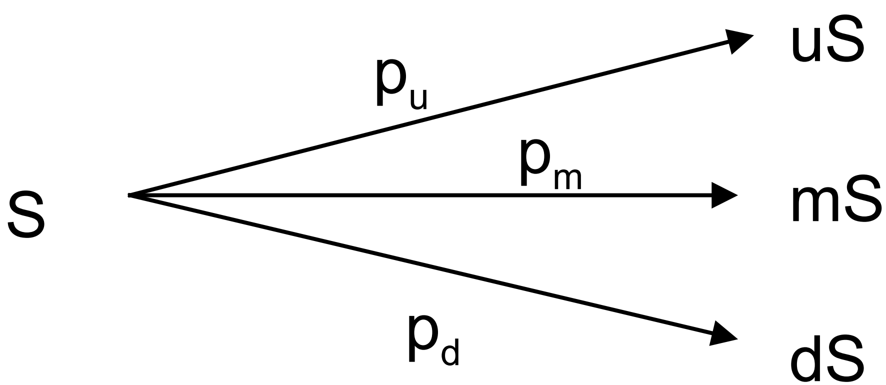
- A simple way to think about it is to collapse two steps of a binomial tree of half size in a single step
- It as a mathematical trick to reduce computation efforts (It can be seen as the **explicit** solution of a PDE)
- With the binomial model, after matching the first and the second moment, we are left with 1 degree of freedom. Now instead we have 3 extra degrees of freedom to specify

Trinomial Model



$$p_u + p_m + p_d = 1$$

- Defining the multiplicative factors *u*, *m* and *d*

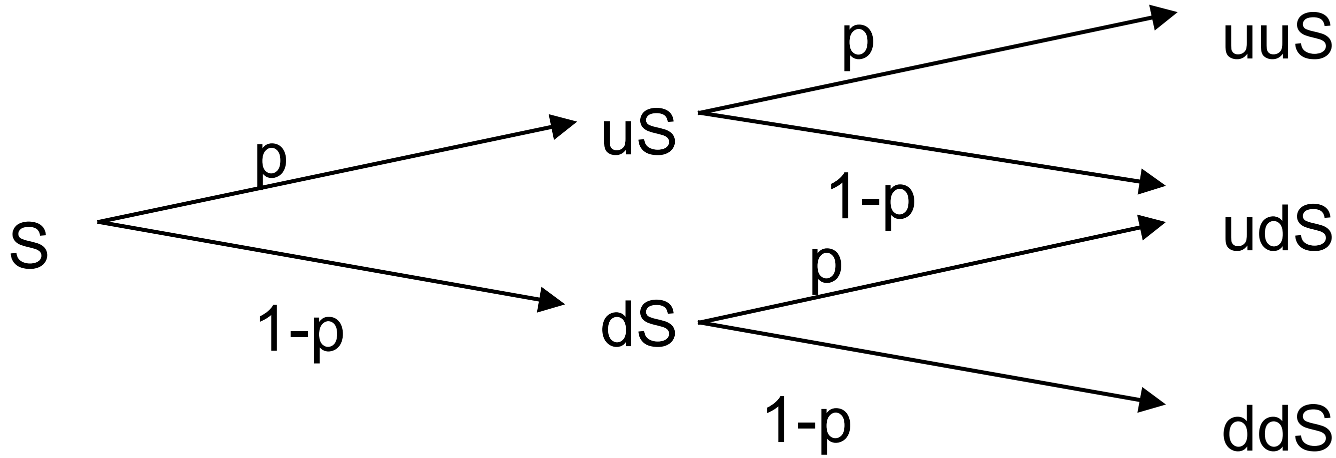


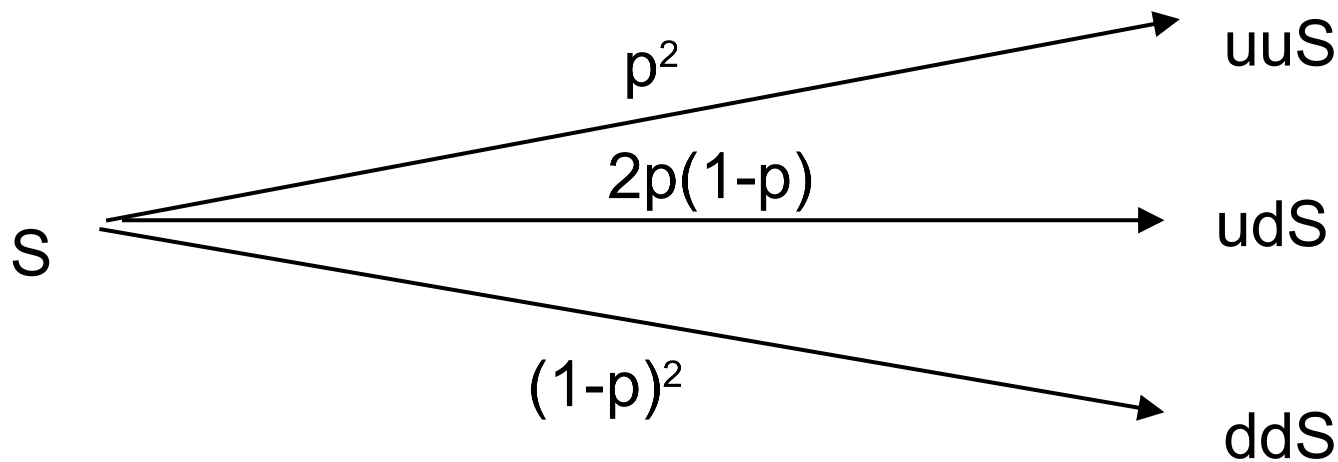
note that there are
5 free parameters

Trinomial Model

- Common ways to fill the extra degrees of freedom are:
  - Combine two steps of binomial tree (CRR, JR, or else …)
  - Match higher order moments
  - Boyle ’86 ($m=1$, $ud=1$, $u=\exp(\sigma(2\Delta t)^{0.5})$)
  - $m=1$, $ud=1$, $p_m=1/3$

Combine 2 Binomial Steps





Trinomial Model Parameters

$$d = 1/u, \qquad p_m = 1/3, \qquad m = 1, \qquad p_m + p_u + p_d = 1$$

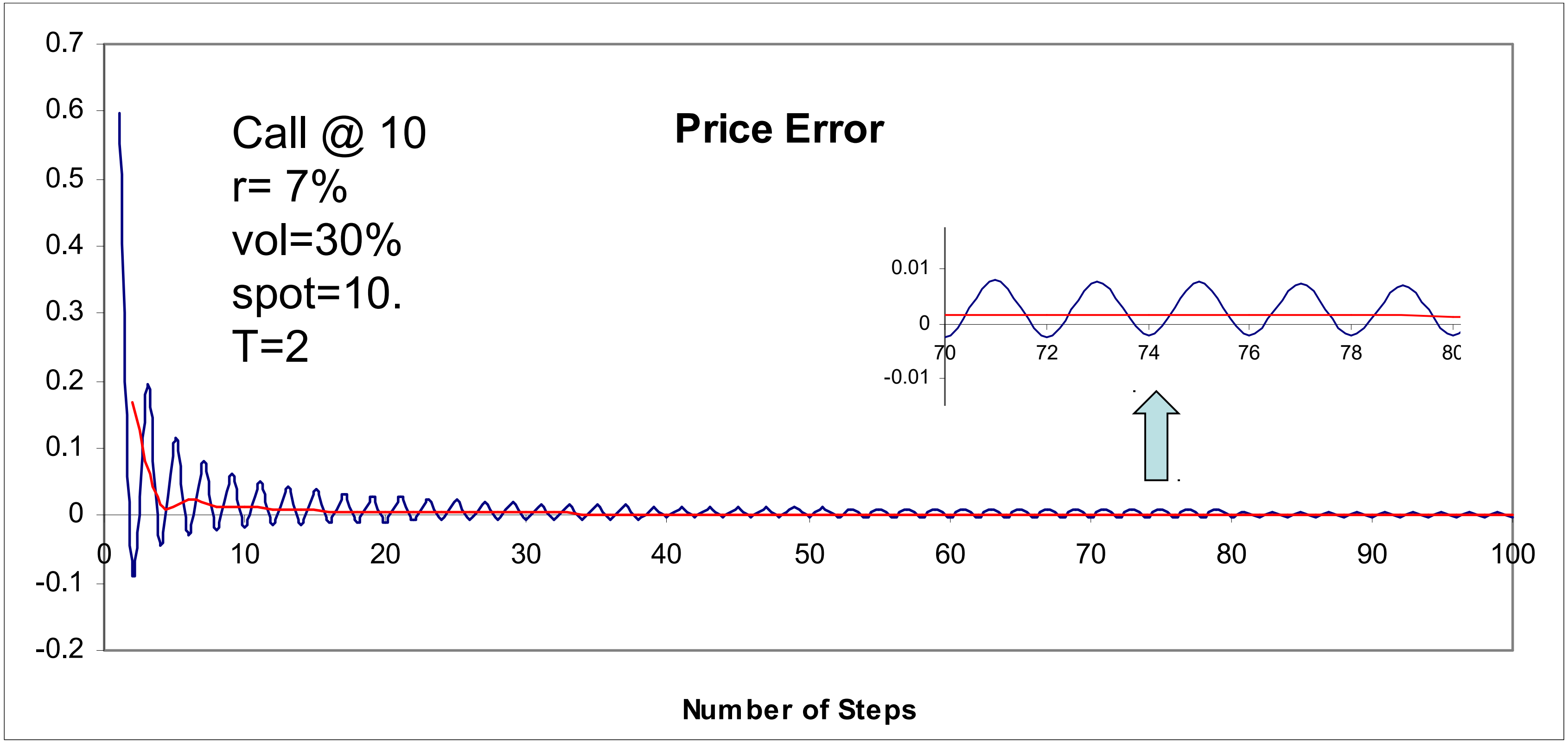
$$p_u u + p_m m + p_d d = \chi_1 \qquad p_u = \frac{\chi_1 - \frac{1}{3} - \frac{2}{3}d}{u-d} = \frac{\xi_1 - \frac{2}{3}d}{u-d}$$

$$p_u u^2 + p_m m^2 + p_d d^2 = \chi_2 \qquad p_u = \frac{\chi_2 - \frac{1}{3} - \frac{2}{3}d^2}{u^2-d^2} = \frac{\xi_2 - \frac{2}{3}d^2}{u^2-d^2}$$

$$\left(\xi_1(u+d) - \frac{2}{3}d(u+d)\right) - \xi_2 + \frac{2}{3}d^2 = 0 \qquad \xi_1 d^2 - \left(\frac{2}{3} + \xi_2\right)d + \xi_1 = 0$$

$$d = \frac{\chi_2 + \frac{1}{3} - \sqrt{\left(\chi_2 + \frac{1}{3}\right)^2 - 4\left(\chi_1 - \frac{1}{3}\right)^2}}{2\left(\chi_1 - \frac{1}{3}\right)}$$

Trinomial (du=1, p_m=1/3, m=1) vs Binomial CRR Example



- It converges faster and without oscillations

Computation Cost

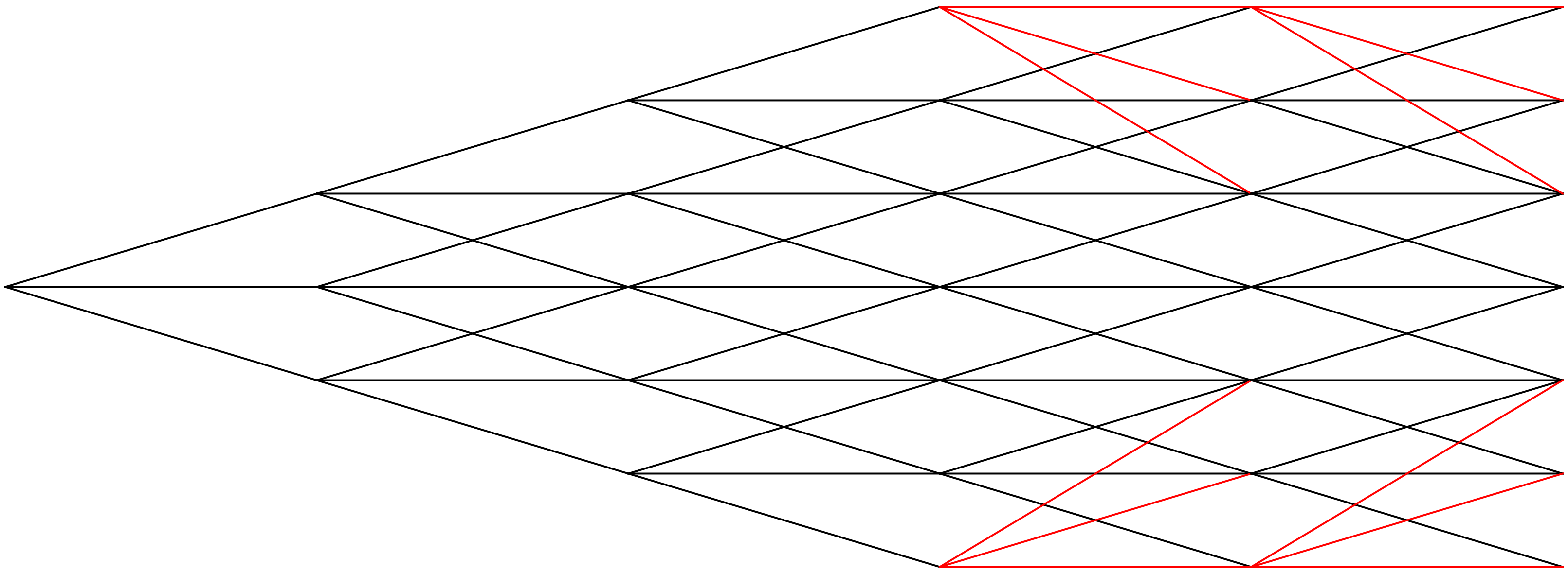
- The computational cost is proportional to the number of nodes in the tree
- Because the number of nodes in the tree depends on N with the relation
  - $NumNodes = 1+3+5+\ldots+(2N+1)$
    $= (2N+1)*(2N+2)/2-2*N(N+1)/2$
    $= N^2+2N+1$
- We can say that the computational cost in terms of number of time step is $O(N^2)$, which is more expensive than a binomial tree with N nodes $O(N^2/2)$, but cheaper than the equivalent binomial tree with 2N nodes $O(2N^2)$

Tree for Ornstein Uhlenbeck

- So far we have only considered trees which match a Geometric Brownian Motion, or its log equivalent, which follow an Arithmetic Brownian Motion
- The ABM has the characteristic that the variance grows linearly in time (i.e. $\sigma^2 t$)
- Tree can also be used to describe processes with bounded variance like Ornstein Uhlenbeck (**Hull, White**)
- As we can imagine the number of nodes will stop growing at some point, as it is no longer necessary to expand the range of prices in the tree
- This type of tree is called “truncated tree”

Tree for Ornstein Uhlenbeck

- Hull White Tree
  - The tree is truncated from the fourth time step onward (note the anomalous node connections in red)



Tree for Ornstein Uhlenbeck

- For a Ornstein Uhlenbeck with zero long term mean:

$$Z_{t+\Delta}=e^{-\alpha\cdot\Delta}\cdot\left[Z_t+e^{-\alpha\cdot t}\int_t^{t+\Delta}e^{\alpha\cdot u}dW_u\right]$$

O-H solution

$$E_t\left[Z_{t+\Delta}\right]=e^{-\alpha\cdot\Delta}\cdot Z_t$$

Conditional mean

$$Var_t\left[Z_{t+\Delta}\right]=\frac{1}{2\cdot\alpha}\cdot\left(1-e^{-2\cdot\alpha\cdot\Delta}\right)$$

Conditional variance (note that it does not depend on Z_t)

- and for convenience we set $M=e^{-\alpha\Delta}-1,\quad V=Var_t\left[Z_{t+\Delta}\right]$

$$E_t\left[Z_{t+\Delta}\right]=\left(M+1\right)Z_t$$
$$E_t\left[Z_{t+\Delta}^2\right]=V+\left(M+1\right)^2 Z_t^2$$

Tree for Ornstein Uhlenbeck

$$\begin{cases} p_d + p_m + p_u = 1 \\ p_d Z_{t+\Delta}^{j-1} + p_m Z_{t+\Delta}^{j} + p_u Z_{t+\Delta}^{j+1} = E\left[Z_{t+\Delta} \mid Z_t^j\right] \\ p_d \left(Z_{t+\Delta}^{j-1}\right)^2 + p_m \left(Z_{t+\Delta}^{j}\right)^2 + p_u \left(Z_{t+\Delta}^{j+1}\right)^2 = E\left[\left(Z_{t+\Delta}\right)^2 \mid Z_t^j\right] \end{cases}$$

set of local conditions to be verified at regular nodes

$$\begin{cases} p_d + p_m + p_u = 1 \\ p_d Z_{t+\Delta}^{j-2} + p_m Z_{t+\Delta}^{j-1} + p_u Z_{t+\Delta}^{j} = E\left[Z_{t+\Delta} \mid Z_t^j\right] \\ p_d \left(Z_{t+\Delta}^{j-2}\right)^2 + p_m \left(Z_{t+\Delta}^{j-1}\right)^2 + p_u \left(Z_{t+\Delta}^{j}\right)^2 = E\left[\left(Z_{t+\Delta}\right)^2 \mid Z_t^j\right] \end{cases}$$

set of local conditions to be verified at truncated upper nodes

$$\begin{cases} p_d + p_m + p_u = 1 \\ p_d Z_{t+\Delta}^{j} + p_m Z_{t+\Delta}^{j+1} + p_u Z_{t+\Delta}^{j+2} = E\left[Z_{t+\Delta} \mid Z_t^j\right] \\ p_d \left(Z_{t+\Delta}^{j}\right)^2 + p_m \left(Z_{t+\Delta}^{j+1}\right)^2 + p_u \left(Z_{t+\Delta}^{j+2}\right)^2 = E\left[\left(Z_{t+\Delta}\right)^2 \mid Z_t^j\right] \end{cases}$$

set of local conditions to be verified at truncated lower nodes

Tree for Ornstein Uhlenbeck

- Imposing the additional arbitrary conditions:

$Z_i^j = j\,\Sigma$ where -i <= j < = i equi-spaced nodes

$\Sigma = \sqrt{3V}$ reproduce the kurtosis of a normal distribution at the first node

- we obtain the following probabilities

	regular node	upper node	lowernode
$p_u =$	$\frac{1}{6} + \frac{j^2 M^2 + jM}{2}$	$\frac{7}{6} + \frac{j^2 M^2 + 3\,jM}{2}$	$\frac{1}{6} + \frac{j^2 M^2 - jM}{2}$
$p_m =$	$\frac{2}{3} - j^2 M^2$	$-\frac{1}{3} - j^2 M^2 - 2\,jM$	$-\frac{1}{3} - j^2 M^2 + 2\,jM$
$p_d =$	$\frac{1}{6} + \frac{j^2 M^2 - jM}{2}$	$\frac{1}{6} + \frac{j^2 M^2 + jM}{2}$	$\frac{7}{6} + \frac{j^2 M^2 - 3\,jM}{2}$

Tree for Ornstein Uhlenbeck

- Further, we require that the probabilities are between 0 and 1. This imposes a restriction on how much the tree can grow (i.e. at which point in time we start to truncate the tree).
- At every time step, the maximum and minimum *j* in absolute value must satisfy:

$$|j| \leq j_{\max} = -\frac{0.1835}{M}$$

Tree for Ornstein Uhlenbeck

- The construction procedure described applies to a O-H with zero long term mean and null initial condition $Z_0=0$.
- If any of these conditions is not verified, the construction happens in two steps
  - Construct a tree centered in zero and mean reverting to zero
  - Shift every time slice of the tree of a constant increment, so that the expected value seen from time t=0 matches the one of the real tree
- Can you prove why this holds? (hint: this is merely a choice of state variable and reconstruction)