Algorithms behind Term Structure Models of Interest Rates I. Valuation and Hedging of Interest Rates Derivatives with the Ho-Lee Model

In this article we implement the well-known Ho-Lee Model of the term structure of interest rates and describe the algorithm behind this model. After a brief discussion of interest rates and bonds we construct a binomial tree and show how to replicate any fixed income type security. This allows us to value any interest rate contingent claim by means of the replicating portfolio. We also discuss the problem of negative interest rates arising in this model and show how to calibrate the model to an observed set of bond prices.

by Markus Leippold[a] and Zvi Wiener[b]

INTRODUCTION

Interest rate risk plays a crucial role in the financial theory. It belongs to the most complex fields in mathematical finance. In this paper, we present a simple interest rate model, the Ho-Lee model. This model appeared in 1986, it is the first term structure model, which allows the matching of the initial term structure. This means that the theoretical zero bond prices are the same as the market prices at the initial date. The Ho-Lee model further builds the basis for more complicated, but more flexible models like Hull-White (1990) and Heath-Jarrow-Morton (1992) which we will present in some later articles. The basic idea of Ho-Lee is to model the uncertain behavior of the term structure as a whole. This is in contrast to the short rate approach to interest rate modeling, where the state variable (in this case the short rate) is represented by a single point on the term structure. The Ho-Lee model can be interpreted as an equivalent of the Cox-Ross-Rubinstein (1979) model for stock options applied to the valuation of interest rate contingent claims. However, in contrast to the real-valued stock price process, Ho-Lee take the class of all functions on R^+ as the state-space of their model. Any such function represents a particular shape of the term structure of interest rates. The deformations of the term structure shape is modeled by means of a binomial tree. The use of the term "tree" in this paper follows the terminology of mathematical economics and finance and is totally different from that of graph theory. The trees presented here are highly recombining, which assures a fast running time of our algorithms. At this point we want to emphasize that this paper does not develop a new method but shows how to implement the algorithm behind the Ho-Lee interest rate model. We use a high level programming language Mathematica to demonstrate the algorithms.

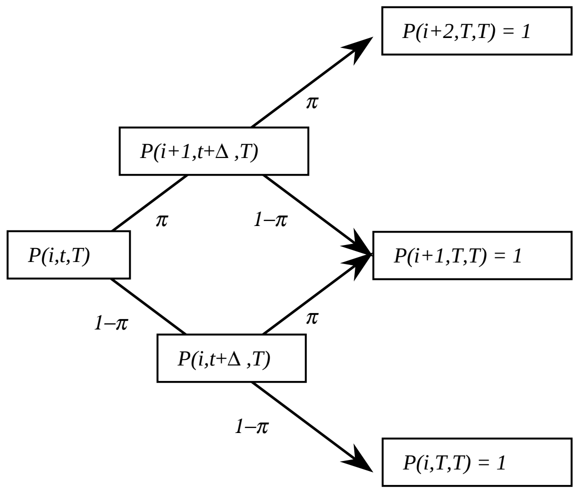
NOTATION AND BASIC ASSUMPTIONS

Before we start building the binomial tree we want to clarify the assumptions on which the Ho/Lee model is built. First there are no market frictions, i.e. we are not considering transaction costs or taxes. Further, all assets are perfectly divisible. Trading takes place at discrete time steps. The market is complete in the sense that there exists for every time T a bond with the respective maturity. For every time t the state-space is finite.

With $P(i,t,T)$ we denote the price of a zero bond in state i at time t, which pays \$1 at the maturity date T. The whole term structure can be captured by the strictly positive function $P(i,t,T)$. We further require the zero bond to satisfy the conditions $P(i,t,t)=1$ and $\lim P(i,t,T)=0$ as $T\to\infty$ for all i and t.

The variable π defines the probability of an upward movement in the binomial tree under the measure **P**. The time steps are set equal to Δ. Next, we introduce the state-independent perturbation function $h^k(t,T)$, which determines the magnitude of the bond price change in the interval $[t-\Delta, t]$. Thus $h^k(t,T)$ denotes the upward move (for $k=u$) or the downward move (for $k=d$) of a bond maturing at time t. The precise formula for the price change appears below. To clarify the notation we plot the evolution of a two-period zero bond.

[a] Markus Leippold, Swiss Institute of Banking and Finance, University of St.Gallen, Merkurstr. 1, 9000 St.Gallen, SWITZERLAND; markus.leippold@unisg.ch.
[b] Zvi Wiener, School of Business, The Hebrew University, Mount Scopus, Jerusalem 91905, ISRAEL; mswiener@mscc.huji.ac.il, http://pluto.huji.ac.il/~mswiener/zvi.html.



Observe that the terminal value of the bond price is equal to one irrespective of the prevailing state. This is in sharp contrast to the stock option pricing trees like the famous Cox-Ross-Rubinstein (1979) model. This feature of bond prices is known as the pull-to-par property, which leads to vanishing volatilities when the time t gets closer to the time-of-maturity T. The appropriate model for this type of stochastic behavior is a Brownian bridge, see Karlin, Taylor (1981).

DERIVATION OF THE MODEL

The Ho-Lee model is actually the simplest arbitrage-free model of interest rates which allows the prefect matching of the initial forward rate curve. The derivation involves three steps:

1. Determine the perturbation function
2. Derive the risk-neutral probabilities
3. Derive the necessary conditions for path-independence in the binomial tree.

The last thing to do is to combine these three steps.

1. Perturbation Function

We know that in a world with no uncertainty, bond prices are related through

$$P(i,t,T) = P(i+1,t,T) = \frac{P(i+1,t-\Delta,T)}{P(i+1,t-\Delta,t)}$$

that holds for all states and all times. We can introduce uncertainty into the model by introducing a perturbation function into the above equation as follows

$$P(i,t,T) = \frac{P(i,t-\Delta,T)}{P(i,t-\Delta,t)} h^d(t,T)$$
$$P(i+1,t,T) = \frac{P(i,t-\Delta,T)}{P(i,t-\Delta,t)} h^u(t,T)$$

Since at maturity the bond price equals its face value, the perturbation functions satisfy the condition

$$h^u(T,T) = h^d(T,T) = 1$$

2. Risk-neutral Probabilities

To derive the risk-neutral probabilities we are using the same arguments as in Cox-Ross-Rubinstein. We take two arbitrary zero bonds with different maturities to construct a portfolio *V*. We invest one unit in the zero bond with time-to maturity *T* and ξ -units in the zero bond with time-to-maturity *S<T*. The value of the portfolio in the upstate *i+1* at time *t< S<T* is

$$V(i+1,t,T,S)=P(i+1,t,T)+\xi P(i+1,t,S)=\frac{P(i,t-\Delta,T)h^u(t,T)+\xi P(i,t-\Delta,S)h^u(t,S)}{P(i,t-\Delta,t)}$$

whereas in the downstate the portfolio has value

$$V(i,t,T,S)=P(i,t,T)+\xi P(i,t,S)=\frac{P(i,t-\Delta,T)h^d(t,T)+\xi P(i,t-\Delta,S)h^d(t,S)}{P(i,t-\Delta,t)}$$

We now want to choose the fraction in the *S*-zero bond such that the return of the portfolio for the period $[t-\Delta,t]$ becomes riskless. Therefore, ξ^* must be such that $V(i+1,t,T,S)=V(i,t,T,S)$. Thus we have

$$\xi^*=\frac{P(i,t-\Delta,T)(h^u(t,T)-h^d(t,T))}{P(i,t-\Delta,S)(h^u(t,S)-h^d(t,S))}$$

To exclude arbitrage opportunities the return on the portfolio $V^*(i,t-\Delta,T,S)$ has to equal the return on the one-period zero bond which equals $1/P(i,t-\Delta,t)$. After some rearranging and substituting ξ^* we arrive at the relationship

$$\pi h^d(t,T)+(1-\pi)h^u(t,T)=1$$

for all *t,T*, $t\le T$. The constant variable π can be interpreted as an equivalent probability measure, more precisely as a so-called risk-neutral probability. It was derived from the no-arbitrage condition for the return on the portfolio *V* which must equal the riskfree return. Since the Ho/Lee model assumes completeness the risk-neutral probability measure is unique and we can price claims as if all investors were risk-neutral.

4. Path-Independence

The third step involves the derivation of the condition for the path independence, which guarantees a recombining binomial tree. This condition can be derived if we look at the zero bond two periods ahead *P(i,t+2*Δ*,T)*. If the tree is recombining, this node can be reached through two paths. Therefore these two paths build a system of two equations, namely

$$P(i+1,t+2\Delta,T)=\frac{P(i,t,T)}{P(i,t,t+2\Delta)}\frac{h^u(t+\Delta,T)h^d(t+2\Delta,T)}{h^u(t+\Delta,t+2\Delta)}$$
$$P(i+1,t+2\Delta,T)=\frac{P(i,t,T)}{P(i,t,t+2\Delta)}\frac{h^d(t+\Delta,T)h^u(t+2\Delta,T)}{h^d(t+\Delta,t+2\Delta)}$$

After solving for the perturbation functions this yields the path-independence condition in the Ho/Lee model as

$$\frac{h^u(t+\Delta,T)h^d(t+2\Delta,T)}{h^u(t+\Delta,t+2\Delta)}=\frac{h^u(t+2\Delta,T)h^d(t+\Delta,T)}{h^d(t+\Delta,t+2\Delta)}$$

To complete the derivation of the model we have to integrate the above three steps. This gives us a difference equation of the first order, which can be easily solved to obtain

$$h^u(t,T)=\frac{1}{\pi+(1-\pi)\delta^{T-t}}$$

and

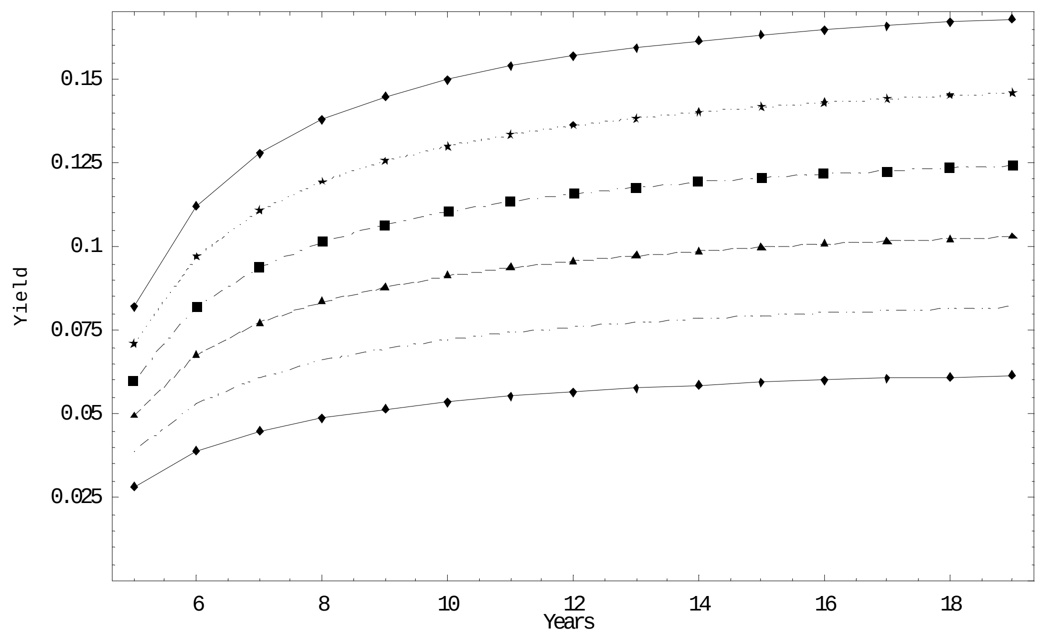
$$h^d(t,T) = \frac{\delta^{T-t}}{\pi + (1-\pi)\delta^{T-t}}$$

Obviously, the perturbation functions depend on the time-*to*-maturity. Solving for δ we obtain

$$\delta = \left(\frac{h^d(t,t+\Delta)}{h^u(t,t+\Delta)} \right)^{1/\Delta}, \quad 0 < \delta < 1$$

These last equations which were derived from the three basic inputs, the perturbation function, the risk-neutral probabilities and the path-independence condition, build the core of the Ho/Lee model.

To avoid trivial cases we chose δ to be in the open unit interval. Interest rate uncertainty would completely vanish for $\delta = 1$. If short rate volatilities are estimated or exogenously given the parameter δ is uniquely determined. This is shown in the next section. Note that from the above equation it gets clear that the Ho/Lee model is only capable of producing a monotone binomial tree for all possible initial term structure shapes, meaning that at every instance *t* the term structure curves are non-crossing. The above graphic visualizes this fact. We plot the possible term structure shapes after 5 time steps. At every of the six nodes a term structure evolves which does not intersect any of the other term structures.



BOND PRICES AND INTEREST RATES

Once we have obtained the expression for the perturbation functions, recursive substitution of the bond price provides a closed-form solution for the bond prices in each node of the binomial tree[1]

[1] Most Mathematica commands are transparent and similar to the standard mathematical notations. For readers less familiar with this language we remind that **`a[[2]]`** means the second element of an array **`a`** (double parenthesis). An example of a function definition is **`f[x_]:=x^2`** which means that a function f with a dummy (underscored) argument **`x`** raises it to the second power. The sign **`:=`** means that this definition will be executed not immediately, but when this function is called. For example to apply this function to a variable **`y`** we use **`f[y]`** (single parenthesis). Note that **`f`** can be also ap-

$$P[i\_, t\_, T\_] := \frac{P[0,0,T]}{P[0,0,t]} \delta^{(T-t)(t/\Delta - i)} \prod_{j=0}^{t/\Delta - 1} \frac{\pi + (1-\pi)\delta^{t-(j+1)\Delta}}{\pi + (1-\pi)\delta^{T-(j+1)\Delta}};$$

It can easily be verified that the one period bond corresponds to

$$P(i,t,t+\Delta) = \frac{P(0,0,t+\Delta)}{P(0,0,t)} \frac{\delta^{(t/\Delta - i)\Delta}}{\pi + (1-\pi)\delta^{t}}$$

Remember that we still have to determine δ. Matching the short rate volatility with the tree geometry can do this. From the definition of the short interest rate we know

```
r[i_,t_]:=-Log[P[i,t,t+Δ]]/Δ
```

The vertical distance between the nodes of the interest rate process is

$$r(i+1,t) - r(i,t) = \ln\delta$$

and therefore constant. The variance of the one-period interest rate can be readily calculated as

$$\sigma_r^2\Delta = Var\left(-\frac{\log P(\cdot,t,t+\Delta)}{\Delta}\right) = \pi(1-\pi)(\ln\delta)^2$$

for all $i = \{0,...,t\}$. Here, σ_r denotes the annual short rate volatility. Hence the variance is neither time-dependent nor state-dependent, but constant. It would be easy to show from the above discussion that the processes of logarithms of zero bond prices are affine transformation of the short rate process.
Whenever the variance of the interest rate process, σ_r, is exogenously determined (by empirical estimation), the term δ becomes

$$\delta := \mathrm{Exp}\left[-\frac{\sigma\Delta^{\frac{1}{2}}}{\sqrt{\pi(1-\pi)}}\right];$$

It is common knowledge that in the Ho/Lee model interest rates follow a Gaussian distribution, which might lead to negative interest rates. Fortunately, we can impose some condition on δ such that interest rates will remain positive up to a finite time *U*. Whenever the interest rate becomes negative, we observe zero bond prices taking values greater than one. Therefore a condition, which has to be satisfied to avoid negative interest rates, is

$$P(i = t/\Delta, t, t+\Delta) = \frac{P(0,0,t+\Delta)}{P(0,0,t)} \frac{1}{\pi + (1-\pi)\delta^{t}} \le 1$$

This leads to

$$\delta \ge \left(\frac{\frac{P(0,0,t+\Delta)}{P(0,0,t)} - \pi}{1-\pi}\right)^{1/t}$$

---

plied to an array raising each element of the array to the second power. A double indexed array is a matrix, e.g. **`b[[1, 2]]`** means the second element of the first row of matrix **`b`**.

constraining the choices of the tree probability π. To determine the lowest allowable value for π we have to stick to numerical methods. The function **ProbCond[U]** evaluates the critical π-value for a given time horizon U:

```
ProbCond[U_] :=
 Module[{p},
  p = Ceiling[10 * p /. FindRoot[
      Exp[- σ Δ^(1/2) / Sqrt[p (1 - p)]] ==
      ((P[0,0,U+Δ]/P[0,0,U] - p)/(1 - p))^(1/U), {p, 0.5, 0, 1}]] / 10.;
 If[p >= 1, Abort[]];
  π = p]
```

The function will calculate the rounded probability, which is allowable in order to guarantee positive interest rates.
The forward rate $f(t,T)$ is defined as the interest rate over the period $T+\Delta$ contracted at time t. Obviously it equals

```
f[i_, t_, T_] := - Log[P[i, t, T + Δ] / P[i, t, T]] / Δ;
```

Futures and forward prices can also be implemented quite easily. The comparison of these quantities might be of some interest. It is well known, that in a stochastic interest framework futures and forward prices are not equal (see Cox-Ingersoll-Ross (1981), Jarrow-Obstfeld (1981), Margrabe (1976)). From the definition of futures and forward prices as well as from using a numerical example, we observe that futures and forward prices are just identical at the last time step of the tree. The forward price is given as

```
Forward[i_,t_,T_,S_]:=  P[i,t,S]/P[i,t,T]
```

The futures price is defined as

$$futures(t,T,S) = E_t\left(P(T,S)\right)$$

where E_t is the expectation operator conditional on time t under the risk-neutral probability measure. The futures price can be programmed as

```
Futures[i_, t_, T_, S_] :=
 Sum[π^j (1 - π)^((T-t)/Δ-j) Binomial[(T - t) / Δ, j] P[i + j, T, S], {j, 0, (T-t)/Δ}];
```

TREE FUNCTIONS
In this section we present several functions, which assign values for bond prices or interest rates at each node of the tree. The function

```
BondTree[T_]:= Table[Table[P[i,t,T],{i,0,t/ Δ}],{t,0,T,Δ}]
```

represents the evolution of the bond price **P[i,t,T]** from time t up to the time-of-maturity T. Similarly the function

```
IntTree[T_]:=Table[Table[r[i,t],{i,0,t/Δ}],{t,0,T, Δ}]
```

evaluates the short interest rate `r[i,t]` at each node in the tree. We also define the tree functions for futures and forwards as

```
ForwardTree[T_,U_]:= Table[Table[Forward[i,t,T,U], {i,0,t/Δ}],{t,0,T, Δ}]
```

and

```
FuturesTree[T_,U_]:= Table[Table[Futures[i,t,T,U],{i,0,t/Δ}],{t,0,T, Δ}]
```

It is useful to have the information on the whole term structure at every node in the tree. The term structure can either be expressed by means of bond prices or by means of forward rates. The function

```
TermStructureTree[T_,tree_]:= Switch[tree,
      bond,Table[ Table[ P[i,t,s],{i,0,t/Δ},{s,t,T,Δ}],{t,0,T,Δ}],
      forw,Table[ Table[ f[i,t,s],{i,0,t/Δ},{s,t,T,Δ}],{t,0,T,Δ}]]
```

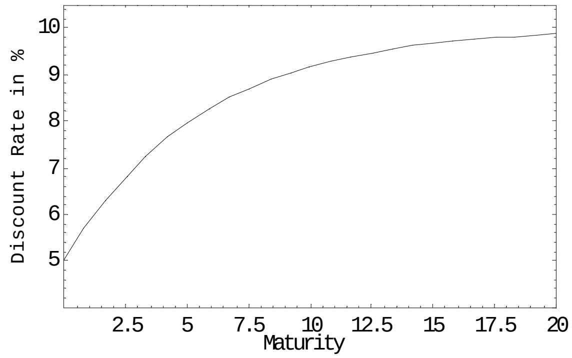
allows to compute these values. If the argument `tree` is set equal to `bond`, the term structure is expressed by means of bond prices. If `tree` is equal `forw` the term structure is expressed by means of forward rates (note that this is similar to the Heath-Jarrow-Morton model with constant forward rate volatility).
Before start calculating an example, we define a function, which will draw the trees for bond prices and short interest rates. The function is called `TreePlot[T,tree]`:

```
TreePlot[T_,tree_,U_:0]:=Module[{AA={},x},
            x = Switch[tree,
                        bond,BondTree[T],
                        int, IntTree[T],
                        for, ForwardTree[T,U],
                        fut, FuturesTree[T,U]
            ];
            Do[ AA = Append[ AA,
                        Table[{ Line[{{j,x[[j,i]]},{j+1,x[[j+1,i]]}}],
                                Line[{{j,x[[j,i]]},{j+1,x[[j+1,i+1]]}}]},{i,1,j}]
                         ], {j,1,Length[x]-1}
            ];
            Show[Graphics[Flatten[AA,2]],Frame->True]
 ]
```

A numerical example
In the following we make the simplifying assumption that the current term structure is given as an exponential function of the form

```
spot[t_] := 0.1 - 0.05 e^(-0.18 t)
```

This creates a reasonable initial term structure for the illustration of how the functions work. The term structure is plotted in the following graph:



Before we can apply the bond price function, we have to make sure that the boundary conditions for the bond price evolution are met. First, the bond prices at time $t=0$ must match the observed term structure, i.e.

```
P[_, 0, T_] := (1 + spot[T])^-T
```

Further, zero bond prices equal *1* at maturity in every state:

```
P[_, t_, t_] = 1;
```

We assume

```
σ = .01;
Δ = 1;
```

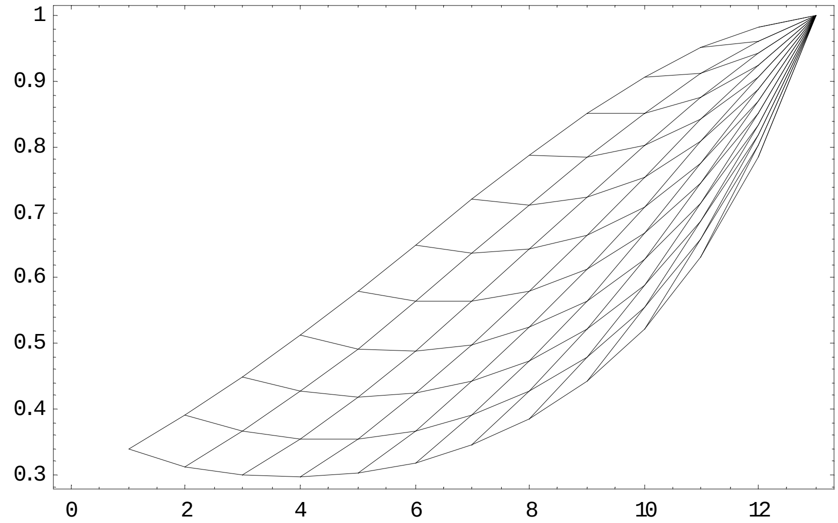
We want to span a tree which does not lead to negative interest rates up to time U=12. Thus we have to evaluate **ProbCond[12]** which gives a critical π-value of 0.6. This choice guarantees the positivity of interest rates at least up to the twelfth step in the binomial tree. Now consider the evolution of a 4-year zero bond. Then the binomial tree for this instrument looks like

```
BondTree[4] // TableForm
```

0.746958				
0.7616	0.809696			
0.806407	0.84001	0.875013		
0.883783	0.902009	0.92061	0.939595	
1	1	1	1	1

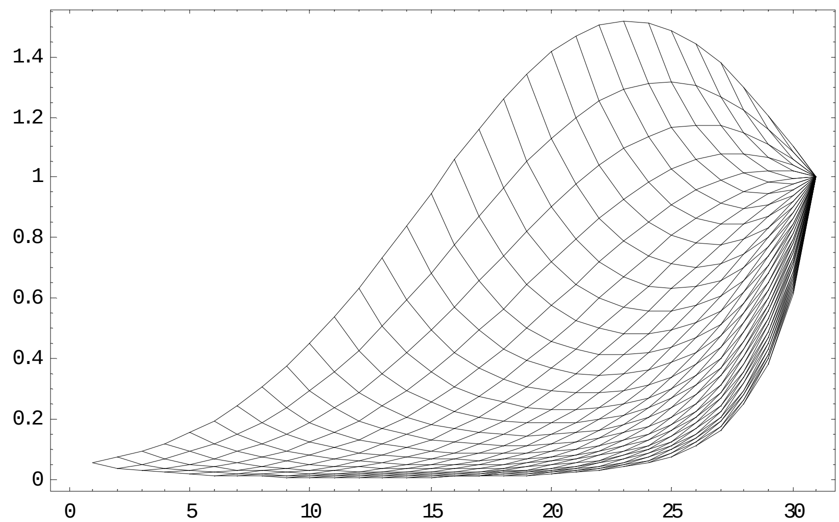
To get an idea how bond prices evolve in a binomial tree, we can use the function **TreePlot[T,tree]**. The graph below shows the evolution of the 12-year bond. A characteristic feature of bond prices is their pull-to-par property. At time T the bond price equals *1* in every of the ($T+1$)-states.

```
TreePlot[12, bond]
```



We can have a look at what happens, if we consider any time-to-maturity *T* which is longer than the critical maturity of *U*. Interest rates will become negative! To obtain an extreme case we assume a time-of-maturity *T*=30. The resulting binomial tree is plotted in the graph below

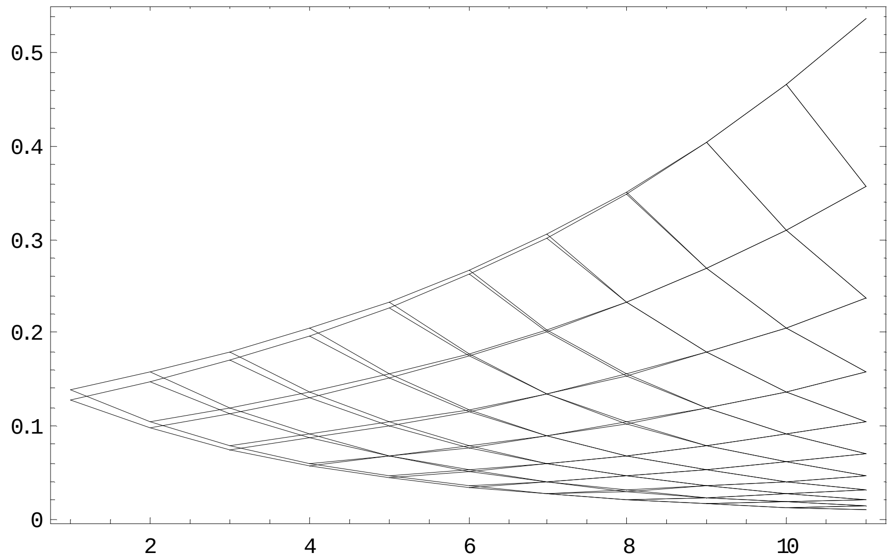
```
TreePlot[30, bond]
```



Clearly, bond prices are in some nodes far above the terminal value of 1. This would clearly give some arbitrage opportunities.

We can visualize the divergence of futures and forward pricing using the `TreePlot` function. Consider the evolution of the 10-year futures and the forward contract on a 30-year zero bond. The following picture shows both trees. The futures contract is substantially lower than the forward contract at time *t*=0.

```
Show[TreePlot[10,for,30], TreePlot[10,fut,30]]
```



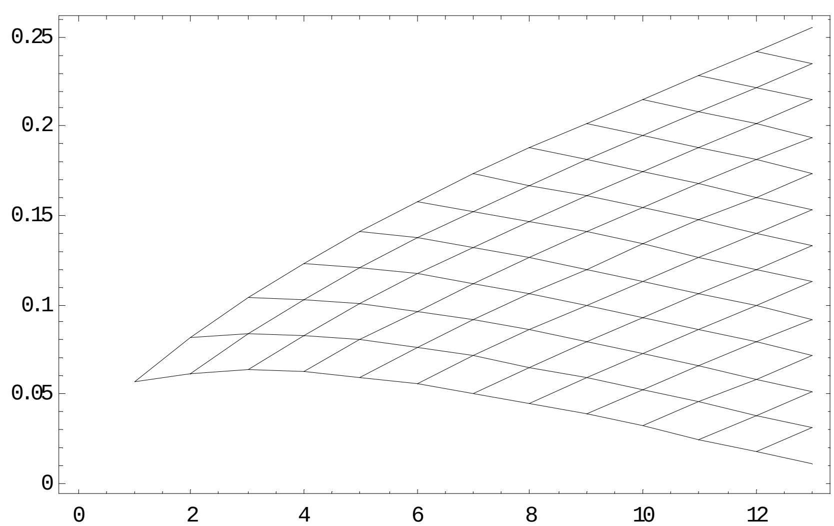
Equivalently to the bond prices, we can calculate the corresponding one-period interest rates in each node.

```
IntTree[3] // TableForm
```

0.0566038			
0.0818613	0.0614489		
0.10392	0.0835076	0.0630952	
0.123544	0.103131	0.0827188	0.0623063

As can be seen from the following graph, the binomial tree for interest rates is equidistant, i.e. the vertical distance between the nodes is constant throughout the binomial tree. Hence, the volatility of the short rate in the Ho-Lee model is constant.

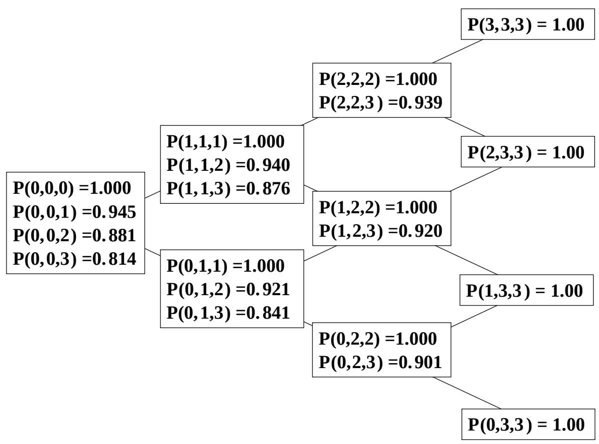
```
TreePlot[12, int]
```



If we are interested in the whole term structure, say e.g. for a 3-year zero bond, we can use the command

```
TermStructureTree[3, bond] // TableForm

1
0.944968
0.881467
0.814327

1           1
0.9214      0.940401
0.840731    0.875764

1           1           1
0.901297    0.919884    0.938854

1           1           1           1
```

to obtain the entire binomial tree for the bond price evolution. The above result is best explained with the following picture where we took rounded values:



The tree thus contains all the information available at every node in the tree.

REPLICATING PORTFOLIO

One of the main insights of modern financial theory is that in a complete market every claim can be replicated by some trading strategy. The cost of the replicating strategy gives the unique price of the claim. Since the Ho-Lee model assumes completeness, we can apply the replication method to derive the prices of arbitrary contingent claims. The replicating trading strategy not only gives the unique price, but also gives the portfolio weights of the replication instruments in each node, by which the claim can be perfectly hedged. The term hedging has different meanings in different contexts. For our purpose here, we will take it to mean the construction of a trading strategy involving two zero bonds of different maturity that replicates the value of our "target" security. This replication approach to hedging, although unrealistic since it assumes the precise execution of the strategy as well as the absence of transaction costs, has nevertheless been shown to be useful in applications.

We illustrate the methodology using the simplest case possible. Suppose we want to price the 2-year zero bond using a replicating portfolio consisting of the 3-year and the 4-year zero bond. The portfolio strategy has to be self-financing. Since the payoff is known at the time-of maturity T=2 (the payoff is 1 in each state), we have to move backwards through the tree. Before we start building the trading strategies, we introduce the following notation. With the two-dimensional vector $(\theta_3(i,t), \theta_4(i,t))$ called "trading strategy" we denote the portfolio weights of the *3*-year and the *4*-year zero bond in the replicating portfolio. Given the assumption of complete markets a two-dimensional trading strategy is enough to replicate any claim in the binomial tree.

To solve our problem we can pursue a two-step procedure:

Step 1: *What must the strategy* $(\theta_3(i,1), \theta_4(i,1))$ *look like at time t=1 to replicate the known payoff of the 2-year zero bond?*

To give an answer to the above question we have to consider state i=1 and state i=0 separately. First, in state i=1 we have

$$\left.\begin{array}{r}\theta_3(1,1)P(2,2,3)+\theta_4(1,1)P(2,2,4)=1\\ \theta_3(1,1)P(1,2,3)+\theta_4(1,1)P(1,2,4)=1\end{array}\right| \Rightarrow \begin{array}{l}\theta_3(1,1)=1\\ \theta_4(1,1)=0\end{array}$$

The portfolio weights allow to calculate the value of the replicating portfolio $v(i,t)$ at time t=1 and state i=1 as

$$\theta_3(1,1)P(1,1,3)+\theta_4(1,1)P(1,1,4)=v(1,1)$$

Analogously, we have for state i=0

$$\left.\begin{array}{r}\theta_3(0,1)P(1,2,3)+\theta_4(0,1)P(1,2,4)=1\\ \theta_3(0,1)P(0,2,3)+\theta_4(0,1)P(0,2,4)=1\end{array}\right| \Rightarrow \begin{array}{l}\theta_3(0,1)\\ \theta_4(0,1)\end{array}$$

which gives rise to a portfolio value corresponding to

$$\theta_3(0,1)P(0,1,3)+\theta_4(0,1)P(0,1,4)=v(0,1)$$

Step 2: *What must the strategy look like at time t=0 to replicate the value of the replicating portfolio* $v(i,t)$ *at time t=1?* Here we have to solve

$$\left.\begin{array}{r}\theta_3(0,0)P(1,1,3)+\theta_4(0,0)P(1,1,4)=v(1,1)\\ \theta_3(0,0)P(0,1,3)+\theta_4(0,0)P(0,1,4)=v(0,1)\end{array}\right| \Rightarrow \begin{array}{l}\theta_3(0,0)\\ \theta_4(0,0)\end{array}$$

The cost of this strategy at time t=0 is thus

$$\theta_3(0,0)P(0,0,3)+\theta_4(0,0)P(0,0,4)=v(0,0)$$

In an arbitrage-free economy, the value $v(0,0)$ must equal the price of the instrument we replicated, i.e.

$$v(0,0)=P(0,0,2)$$

Thus to obtain the price of a financial claim and the weights in the replicating portfolio, which perfectly hedges the claim, we have to work backwards in the tree solving in every node of the tree a two-dimensional equation system. Notice that we illustrated the method using the simplest case, i.e. a bond. However, the method is general enough to value any arbitrary claim on the term structure.

DERIVATIVE PRICING

In this section we present the valuation functions for different derivative instruments. An input for the valuation function is the payoff structure of the instrument under consideration. Therefore, we start by introducing different payoff functions.

The Payoff Functions

The payoff functions give as output the payoff structure of the underlying instrument. Here we present a short list of possible payoffs. Again, we start with the simplest case: the coupon bond.

```
bond[T_, nominal_, coupon_:0]:=
  Append[ Table[ nominal*coupon,{i,0,T/Δ-1},{i+1}],Table[nominal*(1+coupon),
        {T/Δ +1}]
        ];
```

For instance, the coupon bond with nominal value of 1 and coupon of 5% would give

```
bond[3, 1, 0.05] // TableForm
```

```
0.05
0.05     0.05
0.05     0.05     0.05
1.05     1.05     1.05     1.05
```

The payoff of a European option is given by

```
EuropeanOption[K_,T_,optionType_, Underlying_]:= Module[{a},
            a = Switch[ optionType, call, 1,    put, -1,    _, Abort[]];
            Append[Table[0,{i,0,T/Δ-1},{i+1}],
                   Table[ Max[ a*(Underlying - K),0],{i,0,T/Δ}]]
            ];
```

where **K** was used for the strike price, **T** is the expiration date of the option and **Underlying** is the instrument on which the option is written. **optionType** can be used to specify the option's payoff as a **put** or a **call** option. For instance

```
EuropeanOption[K, 2, put, P[i, 2, 3]] // TableForm
```

```
0
0                      0
Max[0, K - P[0, 2, 3]]   Max[0, K - P[1, 2, 3]]   Max[0, K - P[2, 2, 3]]
```

gives the payoff of a two-period European put option on a zero bond with maturity **3**.

Another instrument is the digital option, which pays a fixed amount, say **1** whenever the price of the underlying is above/under the strike price **K**:

```
DigitalOption[K_,T_,optionType_, Underlying_]:= Module[{a,b},
            a = Switch[ optionType,call, 1, put, -1, _, Abort[]];
            b[i_] := Max[ a*(Underlying - K),0];
            Append[ Table[0,{i,0,T/Δ-1},{i+1}],
            Table[ If[b[i]>0 ,1,0],{i,0,T/Δ}]]
];
```

An example is given below

```
DigitalOption[k, 1, call, r[i, 1]] // TableForm
```

```
0
If[Max[0, -k - Log[P[0, 1, 2]]] > 0, 1, 0]   If[Max[0, -k - Log[P[1, 1, 2]]] > 0, 1, 0]
```

is the payoff structure of a one-period digital call option on the spot interest rate. Also of some interest might be the payoff of a forward contract on a zero bond. This is given by

```
ForwardPayOff[T_,S_]:=
      Append[ Table[ 0, {i,0,T-1}, {i+1}],
              Table[ Forward[0,0,T,S] - P[i-1,T,S], {i,1,T/Δ+1}]
]
```

It is obvious that once plugged into the valuation function, this payoff structure must have the present value of zero since the present value of the forward is zero. However as we will see, the valuation formula allows the calculation of the replicating portfolio such that we can perfectly hedge the forward contract.
Basically, there can be constructed arbitrary many European payoff structures. Since we also want to consider the valuation of American options we introduce a function which gives the possible payoff at each node if the option would be exercised:

```
AmericanOption[K_, T_, optionType_, Underlying_]:= Module[{a},
      a = Switch[ optionType,       call, 1,    put, -1,    _, Abort[]];
      Table[ Max[ a*(Underlying  - K),0],{t,0,T/Δ}, {i,0,t}]
];
```

The evaluation of this function for a forward contract on the zero bond *P(U,S)* would give a payoff structure as follows

```
AmericanOption[k, 2, put, F[i, 2, U, S]] // TableFor
```

```
Max[0, k - F[0, 2, U, S]]
Max[0, k - F[0, 2, U, S]]     Max[0, k - F[1, 2, U, S]]
Max[0, k - F[0, 2, U, S]]     Max[0, k - F[1, 2, U, S]]     Max[0, k - F[2, 2, U, S]]
```

The Valuation Functions

In the following we will present the valuation function for European and American options. The valuation function will not only give the arbitrage-free price of the derivative but it will also give the weights of the two bonds with different maturities **S** and **U** in the replicating portfolio at each node. This replicating portfolio perfectly hedges the derivative instrument. Notice that the following functions can be easily modified to value interest rate derivatives instead of bond derivatives.

EuropeanValue[time,S,U,payoffmatrix] prices any claim with non-American features. We denote the maturity of the claim with **time**. The zero bonds used for the construction of the replicating portfolio have maturities **S** and **U**, where **S,U** > **time**. The payoff structure is inferred by the instrument we are considering. The code for the valuation function is

```
EuropeanValue[time_,S_,U_, payoffmatrix__]:=
  Module[{pr,tr,ti,mat, replicmatrix,co},
      pr[T_]:= pr[T]=   Table[{P[i,T,S],P[i,T,U]}, {i,0,T/Δ}];
      tr[T_]:= tr[T]=
                  Table[ {co[T][[i]]*P[i,T,U] - co[T][[i+1]]*P[i-1,T,U],
                         -co[T][[i]]*P[i,T,S] + co[T][[i+1]]*P[i-1,T,S]},
                        {i,1,T/Δ}]/
                  (Table[P[i-1,T,S]*P[i,T,U]-P[i,T,S]*P[i-1,T,U],{i,1,T/Δ}]);
      mat = (Length[payoffmatrix[[-1]]   1)*Δ;
      co[mat]= payoffmatrix[[-1]];
      co[T_]:= co[T]= Apply[Plus,pr[T]*tr[T+Δ],1] + payoffmatrix[[T/Δ+1]];
      {replicmatrix = Table[tr[i],{i,time+Δ,mat,Δ}], co[time] }
      ];
```

Let's consider several examples. Take a coupon bond with nominal value 1, coupon 5% and time-to-maturity of 2. We would like to value this coupon bond using a replicating strategy consisting of the 3-year zero bond and the 5-year zero bond. Thus, we have

```
EuropeanValue[0, 3, 5, bond[2, 1, .05]] // TableFor
```

```
1.82389  -0.751819     1.72224  -0.700001
                       1.6978   -0.670628
1.02279
```

This result has the following interpretation. The value of the coupon bond is 1.02279. The coupon bond can be perfectly hedged at time 0, if we take a long position of 1.82389 in the 3-year zero bond and a short position of 0.751819 in the 5-year zero bond. After one time step, if we find ourselves in state 0, we need to rebalance the replicating portfolio. Now we have to take a long position of 1.72224 in the 3-year zero bond and a short position of 0.70 in the 5-year zero bond. The portfolio weights in state 1 are 1.6978 in the 3-year zero bond and -0.670628 in the 5-year zero bond.

Another example is a European call option with maturity of 2 on a 10-year zero bond with a strike of 0.51. We use the 9-year and the 8-year zero bond to hedge the position in the European option, i.e.

```
EuropeanValue[0,9,8,EuropeanOption[.51,2,call,P[i,2,10]]]//TableForm
```

```
                        0         0
0.390866  -0.35031      0.62664   -0.564663
0.0015997
```

Next we consider a digital call option on the spot rate, time-to-maturity 3 and strike price 10%. We construct the replicating portfolio with the 7-year and the 8-year zero bond:

```
EuropeanValue[0,7,8,DigitalOption[.10,3,call,r[i,3] ]]//TableForm
```

```
                                               7.81961   -7.04501
51.8626  -56.7344     68.7465  -75.5517        107.108   -118.074
                      41.3828  -45.2202        0         0
0.282635
```

Besides these predefined payoff functions, we can also use arbitrary payoff structures. An interesting example is the calculation of state prices. State prices pay 1 unit in state *j*, but nothing in every other state. Thus if we want to calculate today's state price for a claim paying 1 unit at node (*t*=2,*j*=0), we use

```
EuropeanValue[0,3,4,{{0},{1,0}}]//TableForm
```

```
83.2174  -90.2169
0.377987
```

Thus, the price at time 0 for a contract which pays 1 unit if state 0 occurs at time 1 is 0.377987. If state 1 would prevail after one time-step, then we have

```
EuropeanValue[0,3,4,{{0},{0,1}}]//TableForm
```

```
-79.7256  87.6752
0.566981
```

Notice that the sum of the two state prices equals the bond price P(0,0,1) = 0.944968.

Next, we will present the valuation function for the pricing of American call and put options on zero bonds (for an analytic approximation of American options see Barone-Adesi-Whaley (1986)). In contrast to the European case, we have to check in every node whether the option should be exercised immediately or whether the investor should keep the option alive. To calculate the value of American option prices and the corresponding hedging strategy we use:

```
AmericanValue[time_,S_,U_, payoffmatrix_]:=
  Module[{pr,tr,replicmatrix,mat,co},
      pr[T_]:= pr[T] = Table[{P[i,T,S], P[i,T,U]}, {i,0,T/Δ}];
      tr[T_]:= tr[T] = Table[{co[T][[i]]*P[i,T,U]- co[T][[i+1]]*P[i-1,T,U],
                  -co[T][[i]]*P[i,T,S]+ co[T][[i+1]]*P[i-1,T,S]}, {i,1,T/Δ}]/
                  (Table[P[i-1,T,S]*P[i,T,U]-P[i,T,S]*P[i-1,T,U], {i,1,T/Δ}]);
      mat = (Length[payoffmatrix]   1)*Δ;
      co[mat] = payoffmatrix[[-1]];
      co[T_]:=co[T]=
            Max[Transpose[{Apply[Plus,pr[T]*tr[T+Δ],1], payoffmatrix[[T/Δ+1]]}]];
      {replicmatrix = Table[tr[i],{i,time+Δ,mat,Δ}], co[time]}
];
```

Consider now for instance an American call option on the 10-year zero bond with time-to-maturity 2, exercise price 0.45. To construct the replicating portfolio, we use the 6-year and the 3-year zero bonds. The time t=0 value of the option is

```
AmericanValue[0,6,3, AmericanOption[.45,2,call,P[i,t,10]]]//TableForm
```

```
1.22829  -0.904379     0.322005  -0.232554
                       1.79892   -1.34591
0.0247119
```

It is common knowledge that it is suboptimal to exercise an American call option on a non-dividend paying instrument prior to the time-of-maturity. Thus the American call price equals the European call price. We can check this by evaluating the corresponding European call price. This time we use the 4-year and 5-year zero bonds, which should not influence the result.

```
EuropeanValue[0,4,5,EuropeanOption[.45,2,call,P[i,2,10]]]//TableForm
```

```
-2.99073  3.31367     -0.773942  0.863051
                      -4.42695   4.87886
0.0247119
```

Thus we confirmed that the early exercise feature of the American call option has value zero. Evaluating

```
TableForm[
 (AmericanValue[0, 6, 4, AmericanOption[.47, 3, put, P[i, t, 9]]]
 - EuropeanValue[0, 6, 4, EuropeanOption[.47, 3, put, P[i, 2, 9]]])]
```

```
-0.825664  0.692601     -0.281212  0.231814     0.411538  -0.333414
                        0          0            0         0
                                                0         0
0.00954507
```

gives the early exercise feature for an American put option on a 9-year zero bond with expiration in two years. Clearly, due to the early exercise premium the price of the European option is much lower than the price of the American option. The values in the replicating portfolio give a "recipe" for constructing synthetically the early exercise premium.
As in the European case we can evaluate the present value of various derivatives, e.g. an American put option on the forward rate *f(2,3)* with strike 9%:

```
AmericanValue[0,6,3,AmericanOption[.09,2,put,f[i,t,3]]]//TableForm
```

```
0.31456  -0.232792     0.0203784  -0.0147174
                       0.462898   -0.348305
0.0053636
```

Finally, we can evaluate the price as well as the hedging portfolio for any arbitrary payoff structure with American exercise feature:

```
AmericanValue[0, 6, 10, {{0}, {0, 1}, {1, 0, 1}}] // TableForm
```

```
-14.6042  23.4687     27.4495   -41.4904
                      -23.1243  34.9528
0.706808
```

MONEY MARKET ACCOUNT AND FORWARD-NEUTRAL PROBABILITY MEASURES

In this section we will show how the binomial tree, originally constructed by means of the risk-neutral probability measure **P**, can be transformed into a binomial tree that is spanned under the *T*-forward-neutral[2] probability measure which we denote by $\mathbf{P}_T$. The construction of the forward-neutral measure is built in the idea that calculations of interest rate derivatives in a stochastic interest rate framework can be drastically simplified when the bond price process is taken as numéraire since bond prices can be readily observed in the markets. However, it should be stressed that from the viewpoint of computational speed we do not gain any efficiency because the computation of the forward-neutral measure involves the calculation of the money market account. Although in our binomial tree the interest rate process is a path-independent Markov chain, the process of the money market account B is not path-independent. In general, for a binomial lattice the money market account at time *t* can take $2^{(t-1)/\Delta}$ values. Hence the binomial model constructed in the above sections loses most of its simplifying advantages. Nevertheless, we will present the calculation of the forward-neutral measures to gain some theoretical insight and we further present the algorithms in Mathematica to perform the measure changes.

Remember that the time-*t* value of a claim *V* under the risk-neutral measure π is given as

$$V_t = B_t E_t\left[\frac{1}{B_T} V_T\right]$$

where B_t is the money-market account and E_t is the expectation operator under the risk-neutral measure **P** conditional on the information up to time *t*. Now, the forward-neutral measure is defined as the probability measure $\mathbf{P}_T$, which transforms the time-*t* forward price for delivery of an asset at time *T* a martingale. Stated differently, if *V* is again the price of an arbitrary claim, then

$$V_t = P(t,T) E_t^T [V_T]$$

where E_t^T is the expectation operator under the forward-neutral measure $\mathbf{P}_T$ conditional on the information up to time *t*. To evaluate the probability measure $\mathbf{P}_T$ in terms of **P** we equate the above expressions and we obtain

$$\sum_{\Omega} \mathbf{P}_T(\omega) P(t,T) V_T(\omega) = \sum_{\Omega} \mathbf{P}(\omega) \frac{B_t}{B_T(\omega)} V_T(\omega)$$

where ω is an event in a non-recombining binomial tree and Ω is the corresponding event space[3] up to time *T*. Rearranging the above expression yields the *T*-forward measure in each node for the non- recombining tree

$$\mathbf{P}_T(\omega) = \frac{\mathbf{P}(\omega)}{P(T) B_T(\omega)}$$

where, by abuse of notation, *P*(*T*) stands for *P*(*0,0,T*). Before we go on we have to calculate the money market account. The money market account is an adapted non-decreasing stochastic process. At time *t=0* the value of *B* is by definition equal to \$1 and evolves according to a roll-over investment strategy. At every time step the money market account is invested into the one-period bond *P*(*t,T*). Using this definition of the money market account it is easy to show that its value must correspond to

$$B(t,\omega)^{-1} = P(t) \left(\prod_{i=1}^{t/\Delta - 1} \frac{1}{\pi + (1-\pi)\delta^{i\Delta}} \right) \delta^{S_t(\omega)}$$

where

---

[2] We adopt the terminology "forward-neutral probability measure", which was used in one of the first articles introducing this measure-change methodology (see Geman (1989)).

[3] Note that since Ω is the event space in a non- recombining tree it has 2^T elementary events up to time *T*.

$$S_t(\omega) = \sum_{i=1}^{t/\Delta - 1} (i - s_i(\omega))$$

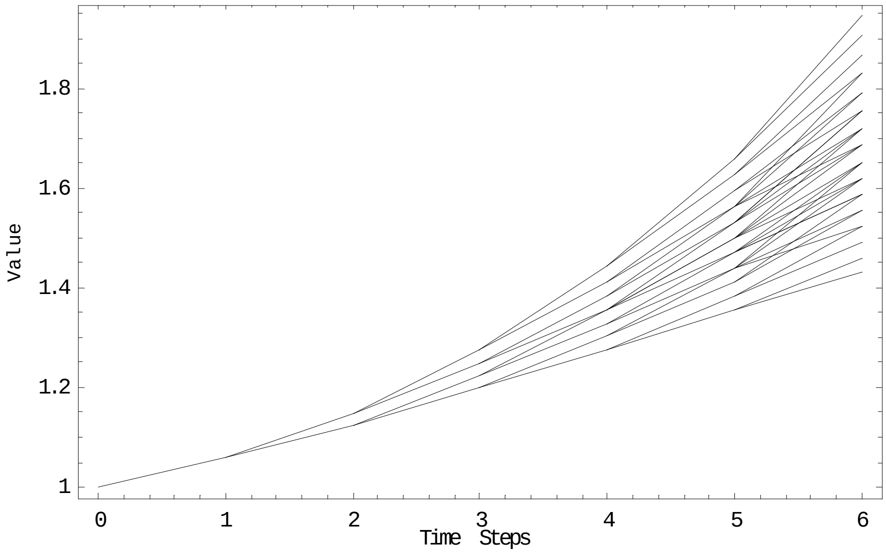
and $s_i(\omega)$ denotes the number of upward movements that have occurred in the history of the event ω up to *i*. We are now able to implement the above expression for the money market account in Mathematica code. First we state the initial conditions for the money market account as

```
treeMM[0] = {1};
treeMM[Δ] = {1/P[0, 0, Δ]};
```

Next we implement the non- recombining binomial tree for the money market account. Its function is given as

```
treeMM[0] = {1};
treeMM[1] = {1/P[0, 0, Δ]};

treeMM[T_] :=
 Module[{n, u, d, b, AA, m},
 n = T/Δ-1;
 AA = {};
Do[
 a = Permutations[Join[Table[d, {n-up}], Table[u, {up}]]];
 u = 1; d = 0;
 b = FoldList[Plus, 0, #]&/@ a;
AA = Append[AA, b],
 {up, 0, n}];
BB = Table[j, {2^n}, {j, 0, n}] - Flatten[AA, 1] ;
 1/( P[0, 0, Δ T] (Product[1/(π + (1-π) δ^((i-1) Δ)), {i, 1, T/Δ}]) δ^Apply[Plus, Reverse[Flatten[AA,1]],1] )
]
```

The following picture shows the evolution of the money market account in the non- recombining Ho-Lee binomial tree



As can be observed, the money market account from time *t=0* up to time *t=1* evolves deterministically, since the one-period interest rate is already known at time *t=0*. In more mathematical terms, the interest rate process is an adapted sequence of predictable random variables.

Above we have calculated the expression for the forward-neutral measure of the non- recombining tree. Equipped with the formula for the money market account, the next step is to calculate the forward measure of the recombining tree. This is done by forward induction. Let $A(j,i)$ be the event that up to time step *j*, *i* upward movements have occurred. In the case of $j=\Delta$ we obviously have

$$\mathbf{P}_{\Delta}(A(1,i)) = \frac{\mathbf{P}(A(1,i))}{P(\Delta)B_{\Delta}(i)} = \mathbf{P}(A(1,i)) = \pi^{i}(1-\pi)^{1-i}$$

The one-period forward-neutral measure corresponds to the risk-neutral measure. Therefore, the risk-neutral measure can be considered as a special case of the forward-neutral measure.

Now we take a look at what is happening in time step 2. Then we have for the 2Δ-forward measure

$$\mathbf{P}_{2\Delta}(A(2,i)) = \left(\pi \cdot \mathbf{P}(A(1,i))b(1,i)\delta^{2-i-1} + (1-\pi)\cdot \mathbf{P}(A(1,i-1))b(1,i-1)\delta^{2-i}\right)h(1)$$

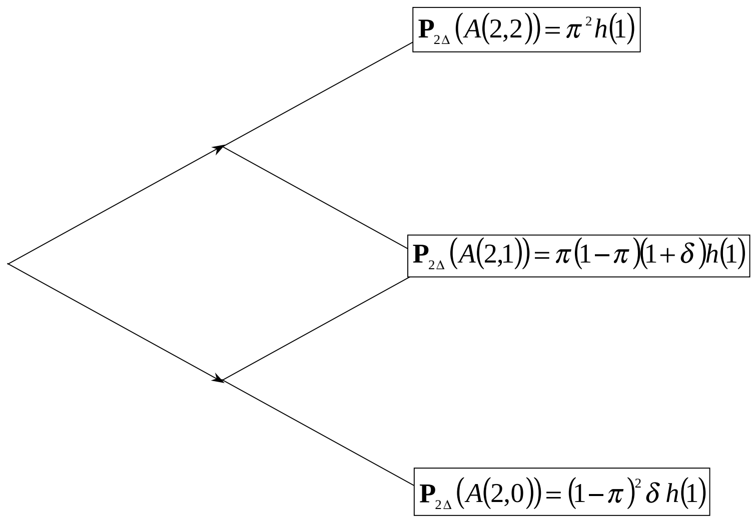
where

$$b(1,i) = \begin{cases} 1 & \text{if } \; i \in \{0,1\} \\ 0 & \text{else} \end{cases}$$

Since $\mathbf{P}(A(1,i))$ is path-independent, we have

$$\mathbf{P}_{2\Delta}(A(2,i)) = \pi^{i}(1-\pi)^{2-i}\left(b(1,i-1)\delta^{2-i} + b(1,i)\delta^{2-i-1}\right)h(1)$$

Consequently, the three forward-neutral probabilities in the recombining tree after two time steps have values as shown in the picture below



One can easily check that

$$\sum_{k=0}^{2} \mathbf{P}_{2\Delta}(A(2,k)) = 1$$

so $\mathbf{P}_{2\Delta}$ is indeed a probability measure.

Assume now that the *T*-forward-neutral measure for time step $j\text{-}\Delta < T/\Delta$ have already been calculated. Then the forward-neutral measure in the node (j,i) is the sum of all forward measures through the paths containing the nodes $A(j\text{-}\Delta, i\text{-}1)$ and $A(j\text{-}\Delta, i)$. Thus we have

$$\mathbf{P}_T(i) = \sum_{S(\omega)=i} \frac{\mathbf{P}(A(j,i))}{P(\Delta)B_\Delta(\omega)}$$

After some algebraic manipulation we derive the *T*-forward-neutral measure in the recombining Ho-Lee binomial tree as

$$\mathbf{P}_T(i) = \left(\prod_{k=1}^{T/\Delta-1} \frac{1}{\pi + (1-\pi)\delta^{k\Delta}} \right) \pi^i (1-\pi)^{T/\Delta - i}\, b(T/\Delta, i)$$

where

$$b(j,i) = \begin{cases} 1 & \text{if} \quad j = i = 0 \\ 0 & \text{if} \quad j < i \ \text{ or } \ i < 0 \\ b(j-1, i-1)\delta^{j-i} + b(j-1, i)\delta^{j-i-1} & \text{else} \end{cases}$$

Observe that in the case of deterministic interest rates we have $\delta = 1$, thus $b(j,i)$ would just equal the binomial coefficient. Thus, in the deterministic interest rate case the *T*-forward measure would coincide with the risk-neutral measure for every arbitrary *T*.

The Mathematica code for the forward measure is given below. First the function $b(j,i)$ is defined as

```
b[0, 0] = 1;
b[m_, s_] :=
 If[m < s || s < 0, 0, b[m - 1, s - 1] δ^(m-s) + b[m - 1, s] δ^(m-s-1)]
```

Finally, the forward measure is given as

$$\text{fp}[\text{i\_}, \text{T\_}] := \left(\prod_{k=1}^{T/\Delta-1} \frac{1}{\pi + (1-\pi)\,\delta^{k*\Delta}}\right) \pi^{i}\,(1-\pi)^{T/\Delta-i}\,\text{b}[\text{T}, \text{i}]$$

SUMMARY

This paper presents an algorithmic approach to the Ho-Lee term structure model. We have provided a simple and hopefully transparent (but not the fastest) algorithm implementing a binomial version of the model. This algorithm allows pricing of any fixed income instruments in the framework of the model, including zero and coupon bonds (both fixed and floating). This binomial tree can be used for hedging as well and we show how this can be done. Hedging in our context is understood as building a replicating portfolio consisting of two basic instruments (zero bonds). The weights of a replicating portfolio have to be adjusted at each node of the tree in order to generate exactly the same payoff as the original security. In addition we provide algorithms of pricing many derivative contracts such as European and American options on bonds and interest rates, digital options and both forward and futures contracts. The code is flexible enough to be extended for the valuation of any arbitrary payoff scheme. Further we show how to make a measure transformation in the Ho-Lee binomial tree. We transform the traditional binomial tree formulated in the risk-neutral measure into a closed-form tree in terms of the forward-neutral measure, which (once calculated) offers some advantages when calculating derivative prices. We use the high-level computer language Mathematica to show the essence of the algorithms behind the model.

REFERENCES

Barone-Adesi, G., and R. Whaley, (1987). "Efficient analytic approximation of American option values", *J. Finance* 42, 301–320.

Brennan, M. J., and E.S. Schwartz, (1977). "The valuation of the American put option", *J. Finance* 32, 449–462.

Brennan, M. J., and E.S. Schwartz, (1979) . "A continuous-time approach to the pricing of bonds", *J. Banking Finance* 3, 133–155.

Cox, J. C., J.E. Ingersoll, and S. Ross, (1981). "The relation between forward prices and futures prices", *J. Financial Economics* 9, 321–346.

Cox, J. C., S. Ross, and M. Rubinstein, (1979). "Option Pricing: A Simplified Approach", *J. Financial Economics* 7, 229–263.

Duffie, D., (1992). "Dynamic Asset Pricing Theory", Princeton University Press, Princeton.

Geman, H., (1989). "The Importance of Forward Neutral Probability in a Stochastic Approach of Interest Rates", *Working Paper*, ESSEC, Paris.

Heath, D., R. Jarrow, and A. Morton, (1992). "Bond pricing and the term structure of interest rates: a new methodology", *Econometrica* 60, 77–105.

Hull, J., and A. White, (1990). "Pricing interest rate derivative securities", *Rev. Financial Studies* 3, 573–592.

Hull, J., and A. White, (1994). “Numerical procedures for implementing term structure models I: single-factor models”, *Journal of Derivatives*, Fall, 7–16.

Jarrow, R., and G. Oldfield, (1981). “Forward contracts and futures contracts”, *J. Financial Economics* 9, 373–382.
Karlin , S., and H. Taylor, A Second Course in Stochastic Processes , Academic Press, 1981.

Margrabe, W., (1976). “A theory of forward and futures prices, preprint”, Wharton School of Business, University of Pennsylvania.

Vetzal, K. R., (1994). “A survey of stochastic continuous time models of the term structure of interest rates”, *Insurance: Mathematics and Economics* 14, 139–161.