Algorithms Behind Term Structure Models of Interest Rates II: The Hull-White Trinomial Tree of Interest Rates

In this article we implement the trinomial tree of the Hull-White model, which can be easily extended to allow different assumptions about the dynamics of the short rate process. We present the Mathematica algorithm for the extended Vasicek and the Black-Karasinski model. Whenever negative interest rates are generated with a positive probability, we make use of alternative branching processes, which guarantee the positivity of interest rates. Finally we show how to price simple options such as caplets, and compare the convergence of trinomial trees with different geometries.

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

INTRODUCTION

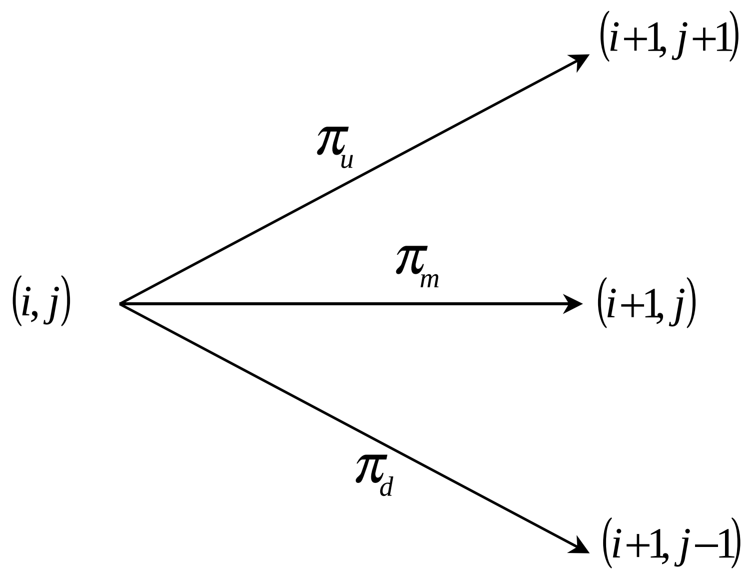
In a previous paper we presented the Ho-Lee (1986) model, the first term structure model, which allows the matching of the initial term structure. Unfortunately, the Ho-Lee model is built on oversimplifying assumptions. The short rate is supposed to follow arithmetic Brownian motion, which does rapidly lead to negative interest rates. We showed how this problem can be controlled by restricting the values of the node probabilities. Nevertheless, positivity of interest rates can be guaranteed only up to a small number of time steps in the binomial tree. Therefore, we can not expect the Ho-Lee model to accurately price derivative instruments, which need a high density of nodes (e.g. barrier options, instruments with frequent cash flows etc.). This paper does not develop a new method but shows how to implement the algorithm behind the Hull-White interest rate model. The Hull-White methodology is used together with analytic power of Mathematica to construct trinomial trees for the generalized Vasicek (Hull-White 1990, 1994) and the Black-Karasinski (1991) interest rate models and a very flexible framework for trinomial trees.

NOTATION AND BASIC ASSUMPTIONS

The Hull-White model is based on the same set of assumptions as the Ho-Lee model. There are no market frictions, no transaction costs nor 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. We assume for every time t the state-space is finite. Denote by $P(i,t,T)$ the price of a zero bond in state i at time t, which pays \$1 at the maturity date T. The entire 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. To abbreviate the notation, the bond prices $P(0,0,T)$ which can be observed from the initial term structure are denoted as $P(T)$

The standard branching process is trinomial. We now need to define the transitions in the tree. With π_u we denote the probability by which the process moves upward from node (i,j) to node $(i+1, j+1)$, with π_m the probability that the process moves to node $(i+1,j)$ and π_d denotes the probability that the process makes a downward move to node $(i+1, j-1)$ (see the figure below).

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



THE DISCRETE MODEL

The trinomial tree is used to construct a discrete time and discrete space Markov approximation of the state variable *x*, which follows the time-inhomogeneous continuous stochastic process

$$dx = (\theta(t) - ax)dt + \sigma(x)dW$$

where *W* is a standard, one-dimensional Brownian motion, $\theta(t)$ is the drift and *a* is the mean reversion. The term $\sigma(t)$ is the volatility term of the diffusion and may depend on the current level of the state variable *x*. The short rate *r* is now assumed to be a function of the state variable *x*. It is well known that in Gaussian interest rate models we can identify the short rate as the state variable (see e.g. El Karoui-Geman-Lacoste (1995)), that is

$$x = r$$

Setting the short rate equal to the state variable results in the generalized Vasicek (1977) model of the term structure where the short rate follows the time-inhomogeneous Ornstein-Uhlenbeck process

$$dr = (\theta(t) - ar)dt + \sigma dW$$

It is obvious that under this selection the interest rates are Gaussian and are thus subject to become negative. To avoid this undesirable property we adopt the methodology proposed by Hull-White (1994) and change the geometry of the tree.

Another way to avoid negative interest rates is to assume a stochastic process for the interest rate which is non-negative almost surely. If we take

$$x = w(r) = \log r$$

the Black-Karasinski (1991) lognormal model with constant volatility is obtained where the short rate follows the stochastic differential equation

$$d \log r = (\theta(t) - a \log r)dt + \sigma dW$$

A well-known criticism of this model is that the specification of the short rate dynamics leads to infinite prices for Eurodollar futures (Hogan and Weintraub (1993)). Again, we can manipulate the tree geometry to introduce an upper bound for the interest rate. Thus, we can guarantee that the interest rate stays within a reasonable range.

Let us define the interest rate as a transformation of the state variable *x*, i.e., as a function $r = w^{-1}(x)$ which we will use in a later section. From now on, we will make the simplifying assumption for the volatility and the mean reversion of being constant, i.e. $\sigma(x) = \sigma$ and $a(t) = a$.

As was shown by Hull-White (1994), instead of directly modeling the discretized version of the stochastic process for *r* (or *x* respectively) it is more efficient to build the trinomial tree in a two-step procedure. We first construct a "level" tree that approximates the process

$$dx = -axdt + \sigma dW$$

Let Δt be the size of a time step and Δx_{ij} be the incremental transition random variable, which is computed using the first order approximation difference equation

$$\Delta x_{ij} = -ax_{ij}\Delta t + \sigma \Delta W_i$$

where ΔW_i is a Gaussian random variable $N\left[0, \sqrt{\Delta t}\right]$. In the trinomial tree the random variable $x_{ij} + \Delta x_{ij}$ is approximated by a discrete random variable on the three nodes $\{x_{i+1,j+a}, x_{i+1,j+b}, x_{i+1,j+c}\}$, where *a*, *b*, and *c* are some positive or negative integers. In the following we will make the assumptions that these three nodes are the neighbors $\{x_{i+1,j-1}, x_{i+1,j}, x_{i+1,j+1}\}$. We note that this imposes a restriction on the tree construction, which is not necessary and can be relaxed if additional conditions are imposed. We will present elsewhere a model where the process does not necessarily evolve to the neighboring nodes.

Before we proceed, we note that the conditional expectation value and the variance of the Gaussian random variable are given as

$$E\left[\Delta x_{ij} \middle| x_{ij}\right] = x_{ij}\left(e^{-a\Delta t} - 1\right)$$

and

$$Var\left[\Delta x_{ij} \middle| x_{ij}\right] = \sigma^2 \frac{1 - e^{-2a\Delta t}}{2a}$$

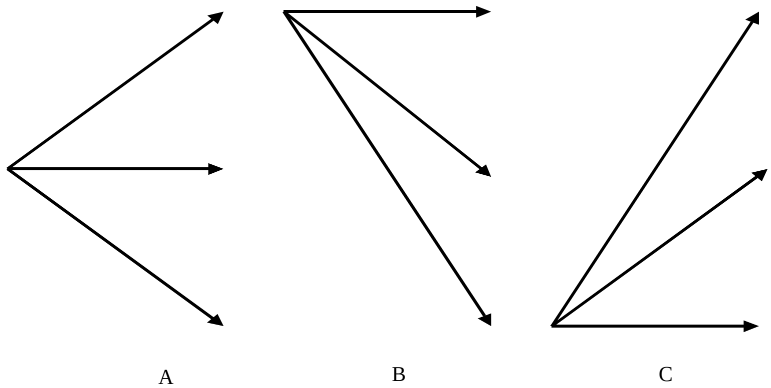
respectively. Often, one takes first order approximations. However, taking exact values gives slightly faster convergence and does not result in lower computational speed.

The construction of the trinomial tree involves matching the expectation value and the variance of *x* at each node in the tree. This leads to a system of equations for the three tree probabilities. We have

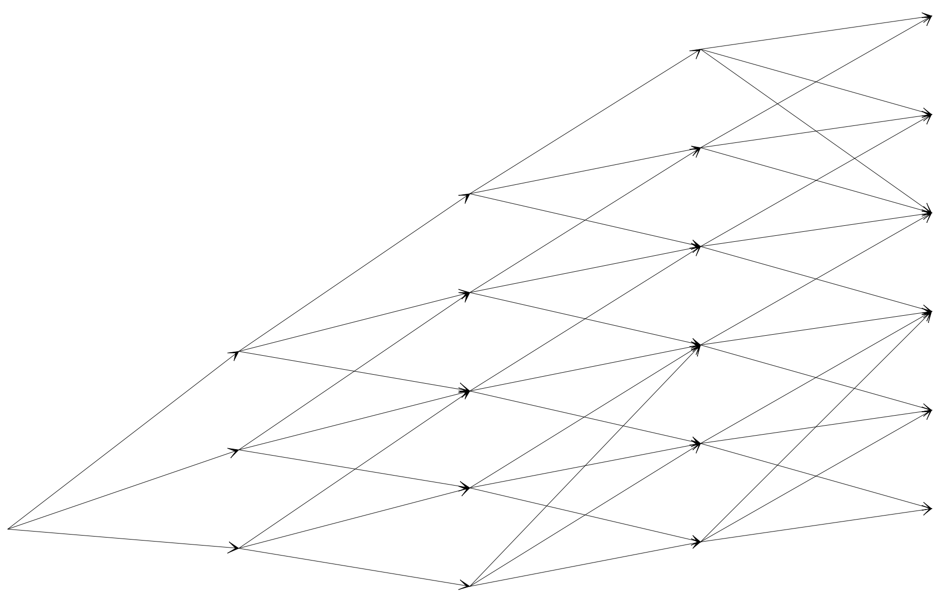
$$\begin{aligned}
&\pi_u + \pi_m + \pi_d = 1 \\
&\pi_u\left(x_{ij} + \Delta x_{ij}\right) + \pi_m x_{ij} + \pi_d\left(x_{ij} - \Delta x_{ij}\right) = x_{ij} e^{-a\Delta t} \\
&\pi_u\left(x_{ij} + \Delta x_{ij}\right)^2 + \pi_m x_{ij}^2 + \pi_d\left(x_{ij} - \Delta x_{ij}\right)^2 - \left(\left(\pi_u - \pi_d\right)\Delta x_{ij} + \left(\pi_u + \pi_m + \pi_d\right)x_{ij}\right)^2 = \sigma^2 \frac{1 - e^{-2a\Delta t}}{2a}
\end{aligned}$$

The determinant of the above system of three linear equations is Vandermonde and equals $2\Delta x_{ij}^3$. In order to guarantee that $\{\pi_u, \pi_m, \pi_d\}$ can be interpreted as probability measures, we have to guarantee the three inequality constraints $\pi_k \geq 0, k \in \{u, m, d\}$. This can now be done in several different ways. First, we can build some constraints on the number of time steps we are considering, as was done e.g. in Leippold-Wiener (1999). However by doing so we impose some severe restrictions on the depth of the tree. This method would probably fail to value either derivatives with complex payoff structures such as barrier options or long term instruments with intermediate payoffs such as claims on mortgage backed securi-

ties. To price these instruments accurately we would require a reasonable depth for the tree. Another possibility is to relax the assumptions that the trinomial tree evolves to the neighbor states as pointed out above. In this case an equation system with six variables and three equality constraints has to be solved. Thus, to uniquely select a particular threesome of transitions, we would have to impose some additional (desirable) constraints. However, in this article we will focus on a third possibility, which is to alter the geometry of the tree such that negative interest rates are avoided. Of course, altering the geometry of the tree is an arbitrary manipulation of the pricing problem and thus subject to some criticism. Nevertheless, it is widely used in practice.
We now introduce three alternative branching processes, which are graphed below



Clearly, the branching process A will be used as long as interest rates are becoming neither negative nor unrealistically high. Whenever in a following step interest rates would become negative, we switch from the branching process A to the branching process C. On the other hand, if interest rates become unrealistically high, we can switch the geometry of the tree from the branching process A to the branching process B. A possible structure of a trinomial tree with the above branching processes would look like in the following picture, where after 2 time steps we start using the branching process C and after 3 time steps we make use of all three branching processes.



The introduction of new branching processes leads to different equation systems, which must be solved. We can let Mathematica do the job.

THE EXTENDED VASICEK MODEL IN CONTINUOUS TIME

Before we start implementing the discrete model in Mathematica, we give a very brief derivation of the continuous (in time and state) version of the extended Vasicek model. We do this not only for the sake of completeness but also to compare the tree methods implemented in Mathematica with the closed-form solutions in a later section.

The continuous-time Hull-White model (1990) is basically an extension of other well-known models, particularly of the models by Vasicek (1977), Cox-Ingersoll-Ross (1981), and Black-Derman-Toy (1990). Hull and White introduced a time-dependent short rate process. The additional degrees of freedom allow matching perfectly the initial term structure of interest rate and volatilities. We restrict ourselves to the extended Vasicek model, since in most pricing problems this model allows the derivation of closed-form solutions. In contrast, the Black-Karasinski model does not allow closed-form solutions. Further, as noted by Hogan-Weintraub (1993) the expectation of the money market account is infinite for any time period.

The extended Vasicek model assumes the drift of the interest rate process to be time-dependent, but not the volatility. Then the process for the short rate follows

$$dr = (\theta(t) - ar)dt + \sigma dW \ ,$$

where W is a standard Brownian motion under the risk-neutral measure. One can solve this stochastic differential equation to obtain

$$r_t = e^{-at} r_0 + e^{-at} \int_0^t e^{as} \theta(s) ds + e^{-at} \sigma \int_0^t e^{as} dW(s) \ .$$

Assuming a sufficiently rich bond market we can express the bond price $P(t,T)$ as

$$P(t,T) = E_t \left[\exp\left(\int_t^T r(s) ds \right) \right],$$

where completeness of the bond market ensures uniqueness of the risk-neutral measure. To solve for an explicit solution for the bond price on can directly calculate the above expectation, which is an easy exercise since the integral of a normal variable is still normal. After some calculations one arrives at

$$P(t,T) = \exp[A(t,T) - B(t,T) r_t],$$

with

$$B(t,T) = \frac{1}{a}\left(1 - e^{-a(T-t)}\right)$$
$$A(t,T) = \frac{1}{2}\sigma^2 \int_t^T B^2(s,T) ds - \int_t^T \theta(s) B(s,T) ds$$

The assumption that interest rates are Gaussian variables allows us to derive closed-form solutions for bond and interest rate options. Using standard change of numéraire techniques one can show that the call price at time t on a bond maturing at S with time-T exercise function

$$X = \left(P(T,S) - K\right)^+ \ ,$$

for $T < S$, is given by

$$\Pi(0,X) = P(0,S)N[d_1] - KP(0,T)N[d_2],$$

where

$$d_1 = \frac{\ln\left(\frac{P(0,S)}{KP(0,T)}\right) + \frac{1}{2}\Sigma_{T,S}^2}{\Sigma_{T,S}}$$
$$d_2 = d_1 - \Sigma_{T,S}$$
$$\Sigma_{T,S} = \left(B(t,S) - B(t,T)\right)^2 \int_t^T \left[\frac{\sigma}{\partial B(t,u)/\partial u}\right]^2 du$$

It is further well known (see e.g. Björk, 1998) that a caplet is a put option on some underlying bond. More precisely, if we assume the cap rate to be r^* which will prevail over a period S-T > 0, then the caplet is equivalent to $1 + r^*(S-T)$ put options on a bond with maturity S, where the exercise date of the option is at T and the exercise price is $\left(1 + r^*(S-T)\right)^{-1}$

The implementation of the caplet formula that we will need in a future section is rather simple. An introduction to Mathematica in Finance can be found in Shaw (1998). First we load the statistics package

```
<< Statistics`Master`
```

The package allows us the define the cumulative distribution function of the standard normal distribution as

```
Φ[z_] := CDF[NormalDistribution[0, 1], z];
```

Then, the formula for the caplet with strike **caprate**, time to maturity **T** and time interval **S-T**, can be typed in as

```
caplet[T_, S_, caprate_] :=
 Module[{v, d, K},
  K = 1/(1+caprate(S-T));
  v = σ/a (1-Exp[-a(S-T)]) Sqrt[(1-Exp[-2aT])/(2a)];
  d = (Log[P[S]/(KP[T])] - 1/2 v^2) / v;
  (KP[T] Φ[-d+v] - P[S] Φ[-d])/K
 ]
```

DEFINING THE PROBABILITIES IN MATHEMATICA

In this section we determine the tree probabilities using Mathematica to solve the corresponding systems of equations. First, we define the value of the transition nodes for each branching process. We denote with **transA** the three node values if branching process A is chosen. Similarly, **transB** and **transC** denote the values if the branching process B and C are selected.

```
transA= {x[j] - Δx, x[j], x[j] + Δx};
transB= {x[j] - 2Δx, x[j] - Δx, x[j]};
transC= {x[j], x[j] + Δx, x[j] + 2Δx};
```

We continue to define two auxiliary functions

```
auxE[x_] := π.x;
auxV[x_] := Plus @@ (auxE[x^2] - auxE[x]^2);
```

which calculate the expectation and variance of the branching processes. Let's define the down-, middle- and up-probabilities by $\{\pi_3, \pi_2, \pi_1\}$, i.e.

```
Unprotect[π]; π = {π3, π2, π1};
```

Since we are starting with a equidistant tree where the starting node is set equal to zero, we can define

```
x[j_] := j Δx
```

The probabilities in the branching process A then solve the equation system

```
probA[j_] = Flatten[ π /. Solve[
{  auxE[transA] == μ[i, j] ,
   auxV[transA] == v[i, j]^2,
   Plus @@ π == 1  },
  π]];
```

The probabilities in the branch B solve

```
probB[j_] = Flatten[π /. Solve[
{  auxE[transB] == μ[i, j] ,
   auxV[transB] == v[i, j]^2,
   Plus @@ π == 1  },
  π]] ;
```

and in branch C

```
probC[j_] = Flatten[ π /. Solve[
{   auxE[transC] == μ[i, j] ,
    auxV[transC] == v[i, j]^2,
    Plus @@ π == 1  },
   π]];
```

The next step is to specify the drift and the volatility function in the above equation systems. As stated earlier, we will use the exact values for the first and second moments

```
μ[i_, j_] := x[j] Exp[-a Δt];
v[i_, j_] := σ Sqrt[(1 - Exp[-2 a Δt])/(2a)] ;
```

The following function is particularly useful for the tree calculations. It produces as output the probability $\pi_z, z = \{-1,0,1\}$ depending on the current level of the node. The values for `jmin` and `jmax` indicate the critical time step where the branching processes C and B replace the standard process A.

```
π[j_, z_, jmin_, jmax_] :=
  Which[jmin < j < jmax,  probA[j] ,
        jmax <= j,        probB[j] ,
        j <= jmin,        probC[j]  ][[z + 2]]
```

PLOTTING THE LEVEL TREE

The tree that has been constructed so far is generic and does not yet reflect the term structure of interest rates or zero bond prices respectively. However, plotting the tree gives some interesting insight on how the geometry of the tree works. We load the two **`packages Statistics`DataManipulation`** and **`LinearAlgebra`MatrixManipulation`** to make use of the functions **`ColumnDrop[]`** and **`Submatrix[]`**. Then the two auxiliary functions

```
auxTree[T_, jmin_, jmax_] :=
 Module[{jtm, jtp, XX},
  jtm = Max[jmin - 1, -T];
  jtp = Min[jmax + 1, T];
   XX = Table[If[ j - i > -2 && j - i < 2, f[T - 1, i], 0],
        {i, jtm + 1, jtp - 1 }, {j, jtm, jtp}];
     Which[
           T <= jmax && T <= -jmin, Return[XX],
           T > jmax  && T > - jmin , Prepend[Append[
                SubMatrix[XX, {2, 2}, {-jmin + jmax - 1, -jmin + jmax + 1}],
                Delete[XX[[-1]], {{1}, {2}}]], Delete[XX[[1]], -{{1}, {2}}]],
           T > jmax, Append[ColumnDrop[
                Transpose[ColumnDrop[Transpose[XX], -1]], -1],
                Delete[XX[[-1]], {1}]],
           T > -jmin, Prepend[ColumnDrop[
                Transpose[ColumnDrop[Transpose[XX], 1]], 1],
                Delete[XX[[1]], {-1}]]]  ]
```

and

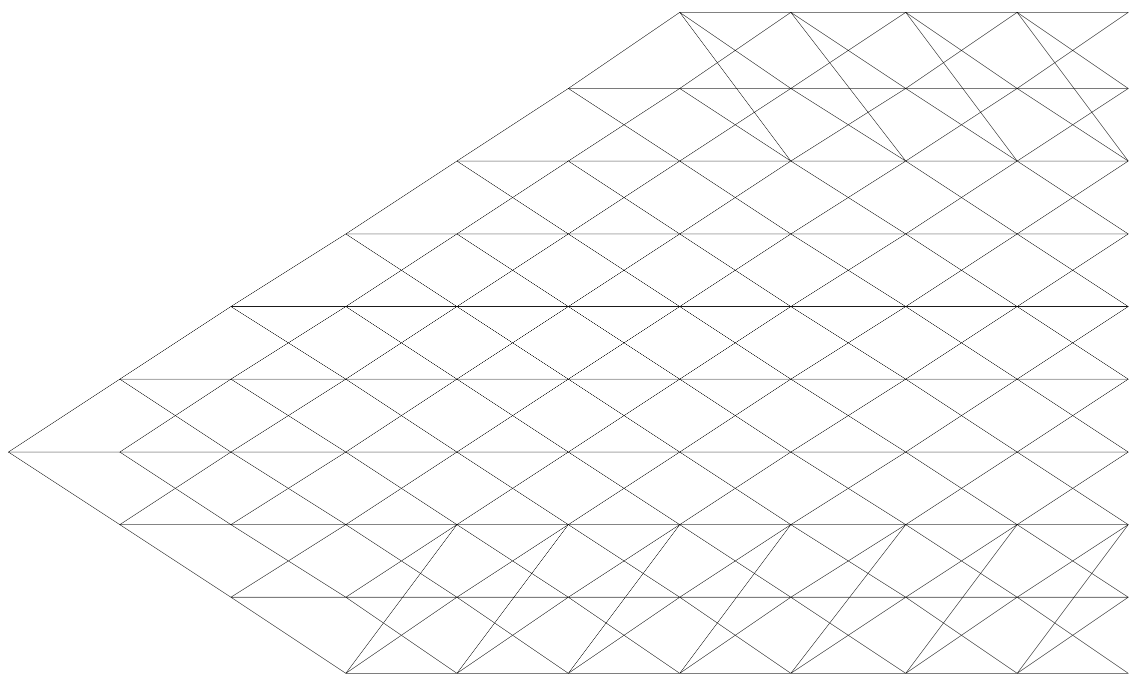
```
precLevel[T_, jmin_, jmax_] :=
 Module[{x, y},
  x = Max[jmin, -T];
  y = Min[jmax, T];
  Table[f[T, k], {k, x, y}]  ]
```

are necessary inputs for the **`TreePlot[]`** function

```
TreePlot[T_, jmin_, jmax_] :=
Module[{LL, FF, HH},
 LL = {};
 Do[ FF = DeleteCases[Transpose[auxTree[u, jmin, jmax]], _0, 2];
  HH = precLevel[u, jmin, jmax];
  f[x_, y_] = {x, y}  ;
  LL =  Append[LL, Table[ Line[{HH[[i]], #}]& /@ FF[[i]], {i, 1, Length[FF]}]];
  Clear[f],
  {u, 1, T} ];
  Show[Graphics[Flatten[LL]]]
 ]
```

As an example we can plot the trinomial level tree with 10 time steps that switches from the branching process A to C after three down-moves and switches from A to B after 6 up-moves:

```
TreePlot[10, -3, 6]
```



CONSTRUCTION OF THE TREE FOR THE STATE PRICES AND THE INTEREST RATES

Now that we have discussed how the nodes of our tree are constructed, we need to define the transitions in the tree. This has to be done in such a way that for each node at time *t* the transitions at time *t+1* are correctly specified. This involves not only finding the transition nodes but also the corresponding probabilities. The function **hh[q,jmin,jmax,a]**, where **q** denotes the time and **jmin** and **jmax** specify at which node we alter the branching process, takes care of this task.

```
hh[q_, jmin_, jmax_, a_] :=
 Module[{jtm, jtp, XX, YY},
  Clear[f];
  XX = {};
  jtm = Max[jmin - 1, -q];
  jtp = Min[jmax + 1, q];
  YY = Table[ If[ j - i > -2 && j - i < 2, z[i - jtm] g[q - 1, i, -i + j, a] , 0],
      {i, jtm + 1, jtp - 1}, {j, jtm, jtp}];
   XX = Append[XX,
     If[ q <= jmax && q <= -jmin, DeleteCases[Transpose[YY], _0, {2}],
        DeleteCases[ Transpose[ Which[
           q > jmax && q > - jmin ,
               Prepend[Append[
                SubMatrix[YY, {2, 2}, {-jmin + jmax - 1, -jmin + jmax + 1}],
                Delete[YY[[-1]], {{1}, {2}}]], Delete[YY[[1]], -{{1}, {2}}]],
           q > jmax,
               Append[ColumnDrop[
                Transpose[ColumnDrop[Transpose[YY], -1]], -1],
                Delete[YY[[-1]], {1}]],
           q > -jmin,
               Prepend[ColumnDrop[
                Transpose[ColumnDrop[Transpose[YY], 1]], 1],
                Delete[YY[[1]], {-1}]]
           ]], _0, {2}]]];
  XX
  ]
```

As already mentioned the function **hh[q,jmin,jmax,a]** finds the correct nodes with the corresponding probabilities. The next step for the construction of the interest rate tree is to convert the generic level tree into the calibrated tree. Here, calibration means that the tree is perfectly matched to the initial term structure. Within the calibrated tree the prices of the zero bonds that mature at each tree time-period coincide with those implied by the yield curve currently observed in the market. Only then our model is arbitrage-free.
Remember, that so far we have constructed a tree which does represent the process

$$\Delta x_{ij} = -a x_{ij} \Delta t + \sigma \Delta W$$

Now, we want the tree to represent the time-inhomogeneous process

$$\Delta x_{ij} = \left(\theta(t) - a x_{ij}\right) \Delta t + \sigma \Delta W$$

where $\theta(t)$ should guarantee the absence of arbitrage. This is achieved by matching the prices of zero bonds implied by the tree to the zero bond prices obtained from the yield curve. Starting with the one period zero bond $P(\Delta t)$ we obtain for $t = \Delta t$ the equality

$$P(\Delta t) = \exp\left[-w^{-1}(\alpha_0)\Delta t\right]\sum_{i=-1}^{1}\pi_i = \sum_{i=-1}^{1} Q_i$$

where α_0 is the shift parameter by which the tree node x_0 is shifted upwards. Q_{ij} is the price of the state *ij*. In other words Q_{ij} is today's value of an instrument, which pays $1 if state *ij* occurs, and nothing in every other state. Generalizing the above relationship we arrive at

$$P((i+1)\Delta t) = \sum_{k=\max[-i,j_{\min}]}^{\min[i,j_{\max}]} Q_{i\Delta,k}\ \exp\left[-w^{-1}(\alpha_{i\Delta t} + k\Delta x)\Delta t\right]$$

In the case when $w^{-1}(x) = x = r$ we can solve the above equation for α_i using

$$\alpha_{i\Delta t} = \frac{1}{\Delta t}\log\sum_{k=\max[-i,j_{\min}]}^{\min[i,j_{\max}]} Q_{i\Delta,k}\ \exp\left[-k\ \Delta r\ \Delta t\right] - \frac{\log P((i+1)\Delta t)}{\Delta t}$$

In the Black-Karasinski model the shift parameter has to be solved numerically, e.g. using the Newton-Raphson method. Once α_i is determined we can calculate the state prices for the next time step using

$$Q_{(i+1)\Delta t,j} = \sum_k Q_{i\Delta,k}\pi_{k,j}\ \exp\left[-w^{-1}(\alpha_{i\Delta t} + k\Delta x)\Delta t\right]$$

where $\pi_{k,j}$ is the probability of moving from node $(i\Delta t, k)$ to node $((i+1)\Delta t, j)$. The Mathematica code which solves this problem for arbitrary functions $w^{-1}(\cdot)$ is **`TreeCalculation[t,jmin,jmax]`**, where **`t`** is the time-dimension of the tree having **`t/Δt`** time steps and **`jmin`** and **`jmax`** are the upper and lower bounds imposed on the level tree.

```
TreeCalculation[t_, jmin_, jmax_] :=
  Module[{ZZ, AA, b, v},
   Clear[QQ, α];
   ΔX = Sqrt[3] σ;
   z[n_] := ZZ[[-1, n]];
   g[i_, j_, z_, m_] :=
     Π[j, z, jmin, jmax] Exp[-ω[(+m⟦-1⟧ + j ΔX)] Δt];
   ZZ = {{1}};
   RR = {{-Log[P[Δt]] / Δt}};
   AA =  w[RR⟦-1⟧];
   Do[
     ZZ =  Append[ZZ, Apply[Plus,
     Flatten[hh[j, jmin, jmax, AA], 1], 1]];
      AA = Append[AA,
        α /. FindRoot[P[(j + 1) Δt] == ZZ⟦-1⟧ .Table[Exp[-ω[(α + h ΔX)] Δt],
            {h, Max[jmin, -j], Min[jmax, j]}] , {α, AA[[-1]]}]],
    {j, 1, t / Δt}];
  Clear[z, g, r];
  QQ = ZZ;
  Do[RR =
     Append[RR, Table[ω[AA⟦i + 1⟧ + j ΔX], {j, Max[-i, jmin], Min[i, jmax]}]],
   {i, 1, Length[AA] - 1}];
  v = 0; If[# < 0, v++]& //@ RR;
  If[v > 0,
   Print["!Warning: You have generated ", v, " negative interest rate(s)\n
          Consider adjusting lower and upper bounds!"]]
]
```

In the code we used the function $\omega(\cdot)$ to denote the inverse of the function $w(\cdot)$.

The output of the function **`TreeCalculation[t,jmin,jmax]`** is stored in the two variables **`QQ`** and **`RR`**, where **`QQ`** denotes the evolution of the state prices in the tree and **`RR`** the evolution of the short rates. Before we want to consider some examples we introduce the function **`IntTreePlot[T,jmin,jmax,list]`** which is equivalent to the function **`TreePlot[t,jmin,jmax]`** which graphs the level tree.

```
IntTreePlot[T_, jmin_, jmax_, list__] :=
Module[{LL, FF, HH},
 LL = {};
 Do[ FF = DeleteCases[Transpose[auxTree[u, jmin, jmax]], _0, 2];
    HH = precLevel[u, jmin, jmax];
    f[x_, y_] := {x, list⟦x + 1, y + Min[-jmin, x] + 1⟧}  ;
     LL =  Append[LL, Table[ Line[{HH⟦i⟧, #}]& /@ FF⟦i⟧, {i, 1, Length[FF]}]];
    Clear[f],
    {u, 1, T / Δt}];
  Show[Graphics[Flatten[LL]] , Frame -> True] ]
```

THE EXTENDED VASICEK MODEL

As stated earlier, in Gaussian models the short interest rate can be set equal to the state variable, i.e. $r = x$. Thus we have

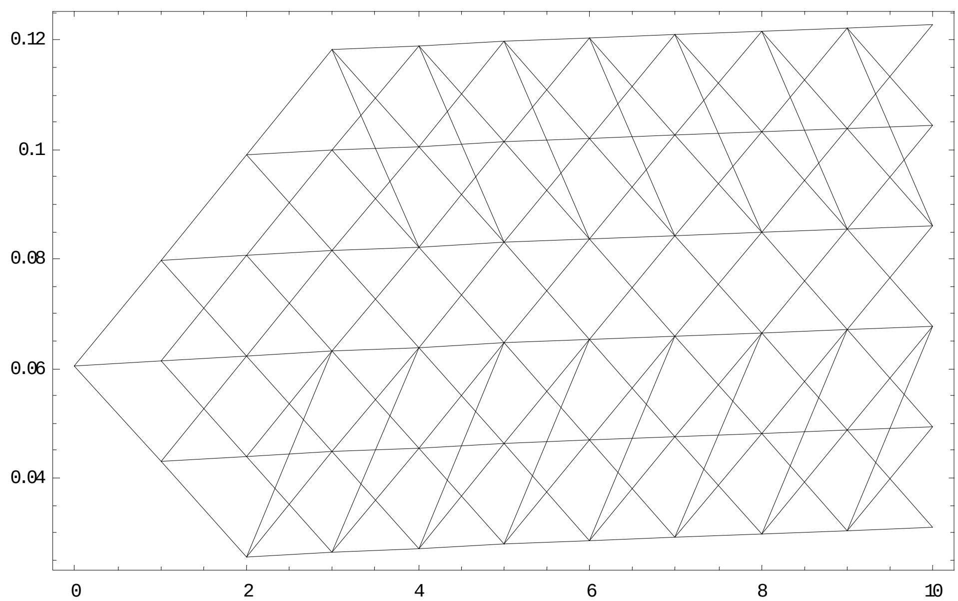
```
w[y_] := y;
ω[y_] := y;
```

To keep things simple we assume that the initial term structure is given as

```
P[t_] := Exp[-(0.07 - 0.01 Exp[-0.1 t]) t];
```

which is a rather flat term structure.
Now consider spanning the tree over 5 years using time steps of 0.5, i.e. we are constructing a trinomial tree with depth of 10. Also we want the standard branching process to change to C after two down-moves and to change to B after 3 up-moves. The rest of the assumptions are given below.

```
a = 0.01; σ = 0.015; Δt = 0.5;
TreeCalculation[5, -2, 3];
IntTreePlot[5, -2, 3, RR]
```



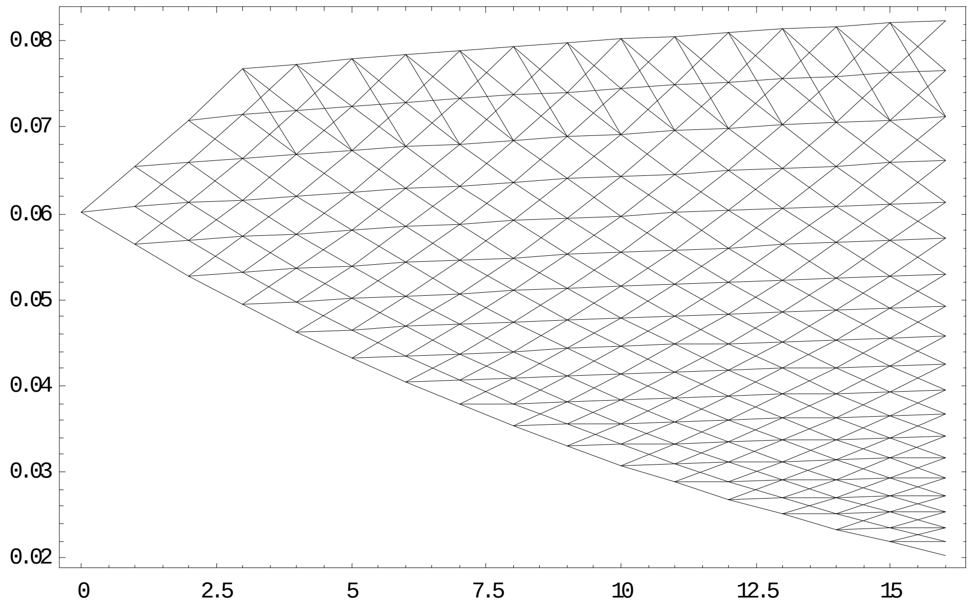
THE BLACK-KARASINSKI MODEL

In the Black-Karasinski model with deterministic volatility the function **w** and its inverse are defined as

```
w[y_] := Log[y];
ω[y_] := Exp[y];
```

For the construction of the tree we use the same initial term structure as in the Vasicek Model. We further make the assumptions that the length of one time period is 0.25. Note that the choice **`w[y_]:= Log[y]`** makes the interest rate a strictly positive process, we do not have to specify a value for **`jmin`** to avoid negative interest rates. Thus in the example below we select a sufficiently high input value for **`jmin`** which does not influence the tree geometry. With the other assumptions as given below the tree takes the following shape

```
a = .02; σ = .085; Δt = 0.25;
TreeCalculation[5, -30, 3]
IntTreePlot[4, -30, 3, RR]
```



FORWARD PROBABILITIES IN THE HULL-WHITE MODEL

Remember that Q_{ij} is the price of the state *ij*, ie, Q_{ij} is today's value of a payoff of $1 in state *ij* and $0 in every other state. Also we know that at every time step *i* the state prices sum up to the observed bond price $P(0, i\Delta t)$. In an arbitrage-free setting $0 \le Q_{ij} \le 1$ for all $i, j \ge 0$ and we also have $0 < P(0, i\Delta t) \le 1$ for all $i \ge 0$. Since the bond price $P(0, i\Delta t)$ is strictly positive we can use it as a numéraire (see Geman-El Karoui-Rochet (1995)). Thus, if we discount the state price Q_{ij} with $P(0, i\Delta t)$, the expression $Q_{ij} / P(0, i\Delta t)$ acts like a probability measure. It is strictly positive and the sum over all states *j* at a time step *i* equals unity.

Consider now an arbitrary claim with payoff-function *X* at time *T*. Then, the time-*t* price of this claim is

$$\Pi(t, X) = P(t, T) E_t^T [X],$$

where $E_t^T[\cdot]$ is the expectation operator under the *T*-forward neutral measure. In our trinomial tree, this means that the price of $\Pi(t, X)$ becomes

$$\Pi(t, X) = \sum_{k=\max[-T/\Delta t,\, j_{\min}]}^{\min[T/\Delta t,\, j_{\max}]} Q_{T,k} \cdot (X)_k$$

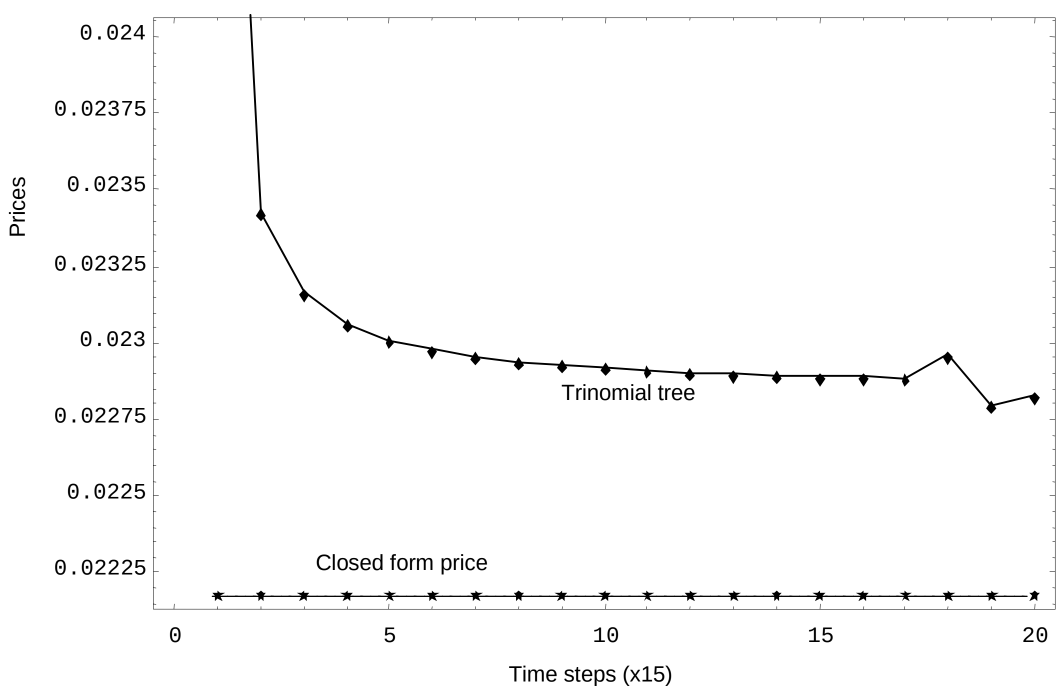
where $(X)_k$ is the payoff if state k occurs at time T.

EXPLORING THE CONVERGENCE FOR THE EXTENDED VASICEK MODEL

In this section we want to take a closer look on how the trinomial tree method for the extended Vasicek model performes relative to the closed form solution. We will consider a single caplet on the interest rate. To simplify things we construct a function, which calculates the values of the caplet both for the numerical method and the closed-form solution. We call this function **caplComp** where **T** is the time-of-maturity of the option, **T/diff** gives the number of time steps, **lower** and **upper** determine at which nodes the geometry of the tree is changes, and **caprate** is the strike of the caplet.

```
caplComp[T_, diff_, lower_, upper_, caprate_] :=
 Module[{res}, Δt = diff;
  TreeCalculation[T, lower, upper];
  res = {(Max[#- caprate, 0]& /@ RR[[-1]]).QQ[[-1]],
  caplet[(T- 1), T, caprate]};
  Clear[Δt, RR, QQ];
  res]
```

We valued a caplet with strike at 4% and time-to-maturity of two years. We manipulated the geometry of the tree in such a way that the interest rate do not become negative. Hovewer, we did not impose any upper bound for the interest rate process. The results can be seen in the picture below.



Clearly, the price calculated with the trinomial tree is higher than the one with calculated with the closed-form solution. This must be due to the fact, that we have capped the trinomial tree in order to prevent the interest rate of becoming negative. This gives more weight to positive interest rates, whereas the closed form solution uses the whole range of the normally distributed interest rates.

SUMMARY

In this article we presented an algorithmic approach to the term structure model of Hull-White. We did not add any new theoretical aspects, but we have shown how to implement such a model into Mathematica. This software seems to be very suitable for solving this type of problems. First, we have to solve some systems of equations to determine the probabilities in different branching processes. Then we have to construct a trinomial tree flexible enough so that one can arbitrarily define the geometry of the tree. Further, within the tree we have to find the roots of an equation numerically in order to determine the level shift in the Black-Karasinski model. All this can be done in the same software package. Finally, we make use of Mathematica's graphical capabilities to explore the convergence of the tree method relative to the closed form solution in the extended Vasicek model.

REFERENCES

Björk, T., (1998). “Arbitrage Theory in Continuous Time”, Oxford University Press.

Black, F.E., E. Derman, and W. Toy, (1990). “A one-factor model of interest rates and its application to treasury bond options”, *Financial Analysts Journal*, 46, 33–39.

Black, F., and P. Karasinski (1991). “Bond option pricing when the short rates are lognormal”, *Finan. Analysts J.*, 47(4), 52–59.

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

El Karoui, N., H. Geman, and V. Lacoste, (1995). “On the role of state variables in interest rate models”, Working paper, Université Paris VI and ESSEC.

Geman, H., N. El Karoui, and R.C. Rochet, (1995). “Changes of numéraire, changes of probability measure, and option pricing”, *Journal of Applied Probability,* 32, 443–458.

Ho T., and S.-B. Lee, (1986). “Term Structure Movements and Pricing Interest Rate Contingent Claims”, *J. Finance*, 41, no. 5, 1011–1029.

Hogan, M., and K., Weintraub (1993). “The log-normal interst rate model and Eurodollar futures”, *Working Paper*, Citybank, New York.

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.

Lieppold, M., and Z. Wiener, (1999). “Algorithms behind Term Structure Models of Interest Rates I. Valuation and Hedging of Interest Rates Derivatives with the Ho-Lee Model”, *Working Paper*, The Hebrew University of Jerusalem.

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

Shaw, W., (1998). “Modelling Financial Derivatives with Mathematica”, *Cambridge University Press*.

Vasicek, O., (1977). “An equilibrium characterization of the term structure”, *J. Financial Economics*, 5, 177–188.