

**Copyright © 2016 RiskSpan, Inc.**

RiskSpan - Black and Karasinski (BK) Model

**Data made beautiful**

This document may be used, modified or copied for internal use, provided the copyright is acknowledged. This is intended to form the basis of your requirements specification. It may not be sold, or used for commercial gain or other purposes without prior written permission.

# Black and Karasinski (BK)

In order to prevent the possibility of negative rates, some prefer to work with what can be described as the lognormal version of the Hull White (HW) model. This alternative can also be defined as a restricted version of the Black and Karasinski (BK) model. The general specification for the BK model has a total of three time-dependent functions, which allows the model to be fit to the zero curve, rate volatility curve, as well as the at-the-money differential rate curve.

*d*In*r* = *[θ∂(t)-a(t*)1n *r]dt + σdz*

Where, ln r = the natural logarithm of the instantaneous short rate, r

q(t) = a time dependent mean reversion level

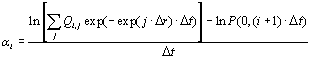
a(t) = a time dependent mean reversion rate

s(t) = time dependent volatility

dz = an increment in a standard Weiner process

If the mean reversion rate and volatility are assumed to be constant, then the resulting simplified model is fit to the zero curve only.

However, because the tree contains values for *x* = ln *r* as opposed to the short rate itself, the displacement factors at each time-step is now calculated as:



To reflect the fact that the short rate at each node must be recovered from the corresponding value of *x*i, j:



## Building Black-Karasinski Trinomial Trees with Changing Time-step

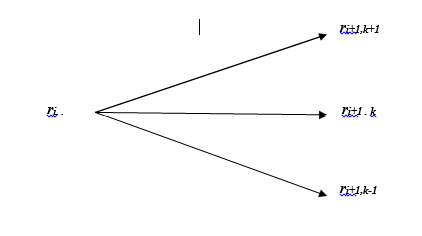
Trinomial trees that are constructed with a constant time-step often do not contain a branch in the tree that coincides with all of the cashflow dates of the underlying instrument. For example, the tree might be built to value a callable bond that pays a semi-annual coupon, with payments that could potentially continue for a ten year period. The tree construction process would be based on splitting the entire period into a large enough number of time-steps so that the resulting bond value is 'sufficiently' accurate. Even if a very large number of steps is chosen, it is unlikely that each coupon payment date will fall exactly on the dates represented by the branches in the tree. There are a number of alternative ways to deal with the problem.

* Discount the cashflow back to the next earliest node. To demonstrate the procedure, suppose that a coupon payment occurs at a date denoted by *r*, which falls between two nodal dates, *t*i and *t*i + 1. The cashflow could be discounted back to time *t*i at the appropriate *r-t*i rates prevailing at all of the nodes at time *t*i. Unfortunately these rates are not directly included in the tree, and must be estimated separately.
* Apportion the cashflow between the adjacent nodes. This approach assumes that a proportion of the cashflow occurs at time *t*i with the remainder paid at time *t*i + 1.
* Change the length of the time-step so that a tree branch always coincides exactly with the coupon payment dates.

Consider a time period between time 0 and time *T* that is divided into a set of finite times denoted as *t*0*, t*1*, t*2*, ......, t*n, where *t*0 *= 0* and *t*n *= T*. Also define ∆*t*i *= t*i + 1 *- t*i as the length of the period beginning at time *t*i. Now when the initial tree is constructed, the state step also becomes time dependent and is calculated as:

****

The tree branching also differs from the constant time-step case. Rather than determine which of the three alternative branching methods are used as a function of the current position in the tree, we now choose the position of the central node as that which is closest to the expected value of the short rate at the end of the period. The branching at all positions in the tree can therefore be depicted as:

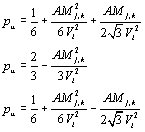


Note that the state subscript *k* is now a parameter to be chosen. Other than that, the tree construction follows the same general steps that are detailed in the previous section. At each node (*i, j*):

1. Determine the value of k such that the central node emanating from the initial node corresponds to a rate equal to the expected value of ri + 1:

E(ri + 1) = ri - a.ri.∆ti. This is computed as the closest integer to E(ri + 1)/∆ri + 1.

1. Compute the probabilities for the top, middle, and bottom branches in the tree. These are determined so that the constructed tree is consistent with the conditional mean and variance of the interest rate process. By also restricting the probabilities to sum to 1, Brigo and Mercurio (2001) show that the probabilities are calculated as:



Equation TemplateWhere,



1. Apply stage two of the tree building process in exactly the same way as for the constant time-step model.

## Building a Lognormal Black-Karasinski Tree

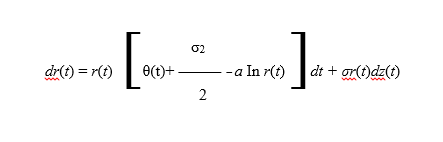
# The procedure outline in the preceding section can be extended to function of the short rate and models with no analytical results, that is, *f(r)* = log *r.* In particular, it is quite applicable to lognormal processes for the short rate. The lognormal HW model or restricted Black- Karasinski (1991) model can be represented as:

*d* In *r(t)* = *a*( In *r* – In **(*t*)*dt* + σ*dz* (i)

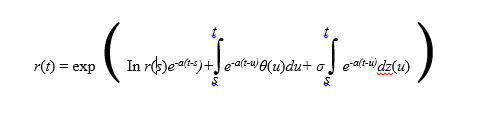
Or written in its more general Black-Karasinski form

*d* In *r*(*t*) = (θ(*t*) - *a* In *r* – *r*(*t*)*dt* + σ*dz* (ii)

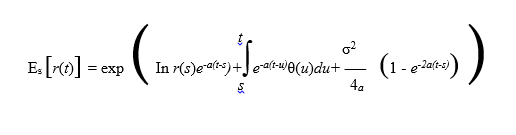
# This lognormal process does not allow negative rates. It is often used by practitioners for this reason. The time-dependent term in the drift is used to calibrate the model to the initial yield curve. The time-homogenous (constant) parameters *a* and *σ* determine the volatility term structure. From (ii) above, we find:



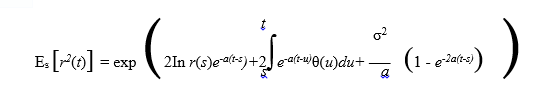
Which has the explicit solution for *s ≤ t* .



Therefore*, r(t)* conditional on ****s is lognormally distributed and has first and



And

We can adapt the Black-Karasinski model to the methodology outlined in the preceding section to construct a short rate tree (for computing discount bond and bond operation that do not have closed-form analytical solution). First, we set *x* = In *r*, yielding the process:

*dx* = (θ(*t*) – *ax*)*dt* +σ*dz*

We then set θ(*t*) = 0, so that the process becomes:

*dx* = - *axdt* +σ*dz*

As before. At time 0, *x* = 0. We assume that

∆*r* = ∆*x=* σ 

The tree is symmetrical with equally spaced state and time steps. The model, however, can be modified to have time-dependent changes in  and ; that is, ∆i = *ti+1* -*ti* and

∆*ri*= *i*

For each *i.* At node (*i, j*), we compute *xi,j = j∆x.* Wethen need to compute the pure security price Q*i,j* at each node (*i , j*) and the displacement (shift) parameters I at each time step *i*= 1, . . . , *N.* The *i* ’s are chosen to correctly price a (i + 1) ∆*t*- maturity discount bond. The ∆*t*- period interest rate at the *j*th node at time *i*∆*t* becomes:

*r i,j =* exp(*i* + *x* *i,j* )

= exp (*i* + *j∆x*) (iii)

Writing bond prices as the sum of discounted pure securities using the short in (iii) yields:

*ni*

*P i+*1 = ∑ Qi,j exp (- exp(*i* + *j∆r*)∆*t*) (iv)

*j=-ni*

Where *ni* denoted the uppermost node at time-step *i* (recall that we don’t know what *ni* isuntil the tree is built \_\_if *j<* then *ni* = *j*). If *i =*0, 0 = In(*r* 0,0 ). We can solve for the i by using Newton-Raphson method. We iterate:

*ni*

∑ Q*i,j* exp (- exp(k+ *j∆r*)∆*t*) – *P*i+j

*j=-ni*

k+1 = k \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

k)

Where

*ni*

∑ Q*i,j* exp (- exp(k+ *j∆r*)∆*t*) exp (k + *j∆r* )∆*t*

∂*Pi+*1

(k) = \_\_\_\_\_\_\_\_\_ =

*j=-ni*

∂*k*

Updating k = k+1 at each time-step until |k+1 -k | <, for arbitrarily small (i.e., 0.0001). Once the tree for *x* is built and we determine the *i*, we can build the tree for *r* by applying (iii) at each node.