This manuscript is program documentation for a model to create a Hull-White trinomial interest rate tree with mean reversion. Although information contained in this manuscript is believed to be accurate, the documentation is offered without warranty, and users agree to assume all responsibilities and consequences from using this documentation.

**General Description**

The Hull-White model (HW) is a trinomial tree of forward short rates, where each rate on the tree can jump to three possible states or nodes on the tree one step later. The median rates on the trinomial illustration are on the horizontal row at the center of the tree, but the numerical values for these median rates generally drift upward from left to right as they approximately equal expected forward rates for the same period. Rates are separated from each other vertically by a constant yield spread used throughout the tree. Therefore, all rates on the tree can be defined as a median rate plus or minus a number of increments above or below that median. This is true both for individual nodes and, importantly, the entire tree. Although forward drift will actually twist the tree, it is easy to see the median rates at the center of the symmetric tree (Figure 1).

**Defining Limits to the Span of the Tree**

The number of increments above or below that median is called the relative row, and the increments are equal up/down between all nodes in a column.

In this implementation, a node on a tree is an element of a two-dimensional array dimensioned with the number of columns in the array equal to the number of horizontal steps on the tree. As described below, a tree contains a limited number of increments above and below the median rate, regardless of how long the tree extends. It is convenient to store individual node values used for pricing in a two-dimensional array. In this implementation, those arrays are dimensioned with positive and negative array pointers. For example, the illustrated tree extends three nodes below and above the median. Such an array would be dimensioned, for example, as X(-3 to 3, 1 to 7). This X array has seven rows and seven columns. For trees used to price longer-maturity instruments, the required tree might have many more columns and the same number of rows. The array includes two triangular areas on the left part of the array that are not used, but, unlike the binomial model¹, most of the elements are used in a trinomial tree represented in a standard rectangular array.

Matching the Tree to Inputs

The rates on the tree must match several inputs. "Sigma" is the annualized standard deviation of rates at a particular step. For example, \( A = 0.01 \) sets the standard deviation at 100 basis points. Since the step size is frequently shorter than one year, this model deannualizes by scaling by the square root of time. For example, a tree containing four steps per year (that is, a “frequency” of four) would use an adjusted \( \sigma \) to calculate the vertical spacing of the tree, called \( \text{dRate}^2 \) here because the change in rate is tied to the standard deviation of short rates:

\[
d\text{Rate} = \sqrt{3} \times \frac{1}{\sqrt{\text{Frequency}}} \times \sigma
\]  

Nodes in general are spaced equally by multiples of \( \text{dRate} \) above and below the median rate. The same \( \text{dRate} \) is used at all steps or columns. However, this implementation supports a short first step, to allow pricing of instruments partway through a regular period.

HW defines a mean-reverting parameter "A." A second variable called "M" adjusts this input for the frequency of the tree:

\[
M = -A \times \Delta \text{time} = -\frac{A}{\text{Frequency}}
\]  

Mean reversion restricts upward and downward movement as the nodes move farther from the median. In particular, this reversion begins at the relative rows \( \text{RowMax}^3 \) (for reversion down toward the median) and \( \text{RowMin}^4 \) (for reversion up toward the median). On the tree illustration, this reversion occurs three levels above and below the median rates. The \( \text{RowMax} \) is calculated by rounding up (that is, finding the next higher integer) following the formula:

\[
\text{RowMax} = \text{Int}\left(0.184 \times \frac{0.184}{M}\right)
\]  

\[
\text{RowMin} = -\text{RowMax}
\]

The integer function in most computer languages truncates a floating-point number down to the nearest whole number less than or equal to the floating-point number. The fraction in the parentheses of Equation 3 is the negative of the HW value (−0.184), and the function as applied rounds down to the next more negative whole number; but, with the sign in Equation 3 outside the INT function, the operation truncates up to the next whole positive number. \( \text{RowMax} \) is always a positive value, because \( M \) is negative; for the same reason, the INT function finds the closest integer value further from the median. The HW tree extends down for the same number of steps as it extends up (Equation 4). When the tree reaches \( \text{RowMax} \) and \( \text{RowMin} \), mean reversion is used.

Probability of Rate Movements

In the interior of the tree, three probabilities define the chance of rising to the next relative row on the next step (\( \text{Pu} \)), moving to a rate on the same relative row one step later (\( \text{Pm} \)), or declining one relative row (\( \text{Pd} \)). Those formulas are shown in Equations 5, 6, and 7:

\[
\text{P}_u = \frac{1}{4} + \frac{1}{4} \text{RowMax} \times M
\]

\[
\text{P}_m = \frac{1}{2} - \frac{1}{4} \text{RowMax} \times M
\]

\[
\text{P}_d = \frac{1}{4} + \frac{1}{4} \text{RowMax} \times M
\]
No mean reversion (middle of the tree):

\[ Pu = \frac{1}{6} + \frac{\text{RelRow}^2M^2 + \text{RelRow} \cdot M}{2} \]  \hspace{1cm} (5)

\[ Pm = \frac{2}{3} - \text{RelRow}^2M^2 \]  \hspace{1cm} (6)

\[ Pd = \frac{1}{6} + \frac{\text{RelRow}^2M^2 - \text{RelRow} \cdot M}{2} \]  \hspace{1cm} (7)

\[ Pu + Pm + Pd = 100\% \]  \hspace{1cm} (8)

At the threshold, \text{RowMax}, this tree stops propagating up. From a node in this upper area, three rates are still possible at the next step. \( Pu \) defines the probability of remaining at the current relative row. \( Pm \) is the probability of declining one row. \( Pd \) is the probability of declining two rows. The formulas for these probabilities are in Equations 9, 10, and 11.

\[ \text{Mean reversion downward (top of the tree)}:\]

\[ Pu = \frac{7}{6} + \frac{\text{RelRow}^2M^2 + 3 \cdot \text{RelRow} \cdot M}{2} \]  \hspace{1cm} (9)

\[ Pm = \frac{1}{3} - \text{RelRow}^2M^2 - 2 \cdot \text{RelRow} \cdot M \]  \hspace{1cm} (10)

\[ Pd = \frac{1}{6} + \frac{\text{RelRow}^2M^2 + \text{RelRow} \cdot M}{2} \]  \hspace{1cm} (11)

\[ Pu + Pm + Pd = 100\% \]  \hspace{1cm} (12)

Below a threshold, \text{RowMin}, the tree stops propagating down. From a node in this area, three rates are possible at the next step. \( Pu \) defines the probability of jumping up two rows. \( Pm \) is the probability jumping up one row. \( Pd \) is the probability of remaining on the same row. These probabilities are shown in Equations 13, 14, and 15:

\[ \text{Mean reversion upward (bottom of the tree)}:\]

\[ Pu = \frac{1}{6} + \frac{\text{RelRow}^2M^2 - \text{RelRow} \cdot M}{2} \]  \hspace{1cm} (13)

\[ Pm = \frac{1}{3} - \text{RelRow}^2M^2 + 2 \cdot \text{RelRow} \cdot M \]  \hspace{1cm} (14)

\[ Pd = \frac{7}{6} + \frac{\text{RelRow}^2M^2 - 3 \cdot \text{RelRow} \cdot M}{2} \]  \hspace{1cm} (15)

\[ Pu + Pm + Pd = 100\% \]  \hspace{1cm} (16)

Although different probabilities apply to different rows of the tree, nodes in the same given row but different columns use the same probabilities. For efficiency, compute all of these probabilities and look up from a table in an array. For example, Table 1 shows probabilities for a particular set of inputs:

---

Table 1

<table>
<thead>
<tr>
<th>Relative Row</th>
<th>Pu</th>
<th>Pm</th>
<th>Pd</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>.8993</td>
<td>.0111</td>
<td>.0896</td>
</tr>
<tr>
<td>1</td>
<td>.1236</td>
<td>.6576</td>
<td>.2188</td>
</tr>
<tr>
<td>0</td>
<td>.1667</td>
<td>.6667</td>
<td>.1667</td>
</tr>
<tr>
<td>-1</td>
<td>.2188</td>
<td>.6576</td>
<td>.1236</td>
</tr>
<tr>
<td>-2</td>
<td>.0896</td>
<td>.0111</td>
<td>.8993</td>
</tr>
</tbody>
</table>

It should be clear from inspecting Figure 1 that for even a modestly short time horizon, a tree could rely on a fairly small table of probabilities over and over. This implementation calculates a table of probabilities in advance and looks up the appropriate probability when working with the tree.

Looking again at the illustration of the tree in Figure 1, notice that going from left to right, exactly three possible destinations are always possible, with probabilities defined above. However, the model determines which nodes (that is, which array elements) to use by following this convention: When branching from a node with a relative row equal to RowMax, the middle destination equals the relative row one row below the starting node. When branching from a node on a relative row equal to RowMin, the middle destination equals the relative row one row above the starting node. All other times, the middle destination equals the relative row of the starting node.

Notice, too, that viewing the linkage of nodes on two contiguous columns is much more complex when viewed from the rightmost column. Some destinations can be arrived at from two, three, four, and even five possible starting nodes. It is much easier to view the linkage from the perspective of the leftmost column in the pair because the patterns are more consistent (as described above).

**Calibrating to the Interest-Rate Term Structure**

Although the forward rates provide a good starting point for the median rates, they are not exactly consistent with no arbitrage. In fact, if the median rate equals the expected value of the short rate, the expected value of the discount factors derived from the tree rates does not match the present value derived from the forward rate. Hull presents a formula for adjusting these median rates to better approximate the median rates consistent with no arbitrage. This implementation instead searches for the right median rates. The algorithm works left to right (forward induction). The algorithm begins with a guess for the no-arbitrage level of the median rate and evaluates a zero-coupon bond, using tree rates derived from the trial median rate and other rates previously calibrated on the tree to the left. The algorithm calculates the error or mismatch between the market price of a zero-coupon bond and the price that follows from a particular choice for the median rate. The median rate is adjusted upward or downward using the standard Newton-Raphson search.

**Search for the No-Arbitrage Median Rate**

First, calculate a zero-coupon bond using the HW probabilities and the tree rates. This valuation relies on previously calibrated rates and the rates derived from a guess for the right median rate for the step in question.

\[
\text{TargetPrc} = \frac{1}{\left(1 + \frac{r}{\text{Frequency}}\right)^n}\quad (17)
\]

\[
\text{TestPrc} = \text{Value of a zero coupon bond maturing at } T = i \\
\text{by backward induction}\quad (18)
\]
Figure 2 illustrates the backward induction process:

\[ Z_{i-1} \] \hspace{1cm} \text{Z is the interim price of a zero-coupon bond}

\[ Z_{i,0}, Z_{i,1} \text{, and } Z_{i-1} \text{ are three bond values at time equal to } i. \text{ If } i \text{ equals the maturity, each of these values equals the face value of the bond. Otherwise, these values would have been derived from other bond prices to the right (not shown). The pricing equals the probability weighted average of the three prices at time equal to } i, \text{ discounted at the rate prevailing at time equal to } i-1. \text{ The price } Z_{i-1,0} \text{ equals the present value of the expected value of } Z_{i-1,1}, Z_{i-1,0} \text{ and } Z_{i-1}.\]

\[ Z_{i,j} = \frac{1}{1 + \frac{r_{i,j}}{\text{Frequency}}} \text{ for each rate (jth rate in column i)} \quad (19) \]

\[ Z_{i-1,j} = \frac{P_U Z_{i,j+1} + P_M Z_{i,j} + P_D Z_{i,j-1}}{1 + \frac{r_{i-1,j}}{\text{Frequency}}} \quad (20) \]

Repeat the process for the remaining nodes, working generally from the right edge of the tree to the left. For nodes on the upper or lower edge subject to mean reversion, be sure to reference the value of the bond at nodes that are one row higher or lower than the pattern illustrated:

For an error defined as the target minus the calculated (Equation 21), the formula for a new guess for the median rate is Equation 22:

\[ \text{PVErr} = \text{Target}\text{-Pre} - \text{Test}\text{-Pre} \quad (21) \]

\[ r_{\text{Median}} = r_{\text{Median}} + \frac{dZ_i}{dr_{\text{Median}}} * \text{PVErr} \quad (22) \]
The Newton-Raphson method requires the derivative of the equation being solved. The price of a zero-coupon bond is a complex weighting of various discount factors. However, only the rightmost set of rates are being adjusted, so the derivative treats everything other than the right-most present-value factor as constants. The derivative of bond price with respect to all of the rates being searched equals the probabilistic sum of the derivatives of the present value.

\[
\frac{dZ_i}{dr_{\text{Median}}} = -\frac{1}{\text{Tenor}} \sum_{j=1}^{\text{Num of Nodes}} \left( \frac{U_{i,j}}{1 + \frac{r_{i,j}}{\text{Frequency}}} \right)^2
\]

(23)

\[
\approx -\frac{1}{\text{Tenor} \left(1 + \frac{r_{i,\text{Median}}}{\text{Frequency}}\right)^2}
\]

(24)

Where \( U_{i,j} \) is the unconditional probability of node \( i, j \)

Note that all rates on the column move up and down in parallel by the same amount, but each node in that step has a different derivative. In practice, an approximation of the derivative equal to the derivative at the median rate provides a close-enough approximation to converge on the arbitrage-free median rate.

**Unconditional Probabilities**

It is fairly straightforward to calculate the unconditional probabilities for each node in a column. This vector of unconditional probabilities is derived from the product of the previous unconditional probability at a node and the HW probabilities, reflecting, of course, the mean reversion pattern required. The logic follows the left-to-right linkage recommended above and, although not immediately intuitive, is simple and terse. The leftmost node is assigned a probability of 100%. Then, each target receives its share of that unconditional probability. If the three jumps are 16.6%, 66.7%, and 16.6%, multiply each marginal probability by the unconditional probability times the unconditional probability of getting to that starting node (100% in this first example). Add these probabilities to the array of unconditional probabilities for the next column, being careful to follow the proper patterns at nodes subject to the mean-reversion patterns.

Repeat for the next column. When there is more than one path that ends at a node, it accumulates probability from each relevant path. As a check, the sum of unconditional probabilities for a column of nodes (that is, for a particular step) should sum to 100%.
About the Author

Stuart McCrory

Stuart McCrory is a trader and portfolio manager who specializes in traditional and alternative investments, quantitative valuation, risk management, and financial software. Before joining BRG, he spent 13 years consulting on a wide range of capital markets issues including litigation consulting, valuation, modeling, and risk management. Previously, he was president of Frontier Asset Management, a market-neutral hedge fund. He held positions with Fenchurch Capital Management as senior options trader and CS First Boston as vice president and market maker, where he traded OTC options and mortgage-backed securities. Prior to that, he was a vice president with the Securities Groups and a portfolio manager with Comerica Bank.

Mr. McCrory is the author of several books, including How to Create and Manage a Hedge Fund: A Professional’s Guide (John Wiley & Sons, 2002).

About Berkeley Research Group

Berkeley Research Group, LLC (www.thinkbrg.com) is a leading global strategic advisory and expert consulting firm that provides independent advice, data analytics, authoritative studies, expert testimony, investigations, and regulatory and dispute consulting to Fortune 500 corporations, financial institutions, government agencies, major law firms, and regulatory bodies around the world. BRG experts and consultants combine intellectual rigor with practical, real-world experience and an in-depth understanding of industries and markets. Their expertise spans economics and finance, data analytics and statistics, and public policy in many of the major sectors of our economy, including healthcare, banking, information technology, energy, construction, and real estate. BRG is headquartered in Emeryville, California, with offices across the United States and in Asia, Australia, Canada, Latin America and the United Kingdom.