MMA708 – Analytical Finance –Jan R. M. Röman

Black-Dermon-Toy Model with Forward Induction

MÄLARDALEN UNIVERSITY

GROUP MEMBERS

Wei Wang

Maierdan Halifu

Yankai Shao

Division of Applied Mathematics

School of Education, Culture and Communication

Mälardalen University

SE-721 23 Västerås, Sweden

Abstract

This report is aimed introduces introduce Black-Derman-Toy one factor model as well as its applications in pricing bonds and interest-rate options. In particular, we focus on how to construct a BDT-Tree to price bonds. Practical and numerical examples are also given in order to make our methods more visible.

Content

Introduction

In finance, the Black-Derman-Toy model is a model of the evolution of the yield curve, sometimes referred to as a short rate model. It is a single stochastic factor (the short rate) determines the future evolution of all interest rates.

The parameters in the BDT model can be calibrated to fit the current term structure of interest rates (yield curve) and volatility structure as derived from implied (from the Black-76 model) prices for interest rate caps. From this point, one can value a variety of more complex interest-rate sensitive securities.

The model was introduced by Fischer Black, Emanuel Derman, and Bill Toy. It was first developed for in-house use by Goldman Sachs in the 1980s and was eventually published in the Financial Analysts Journal in 1990. A personal account of the development of the model is provided in one of the chapters in Emanuel Derman's memoir "My Life as a Quant."

Theoretical Background

Black-Derman-Toy model of interest rates can be used to value any interest rate sensitive security. In explaining how it works, we concentrate on valuing options in Treasury Bonds.

The model has three key features.

1. Its fundamental variable is the short rate—the annualized one-period interest rate. The short rate is the one factor of the model; its changes derive all security prices.
2. The model takes as inputs an array of long rates (yield on zero-coupon Treasury bonds) for various maturities and array of yield volatilities for the same bonds. We call the first array the *yield curve* and the second the *volatility curve*. Together these curves form the *term structure*.
3. The model varies an array of means and an array of volatility changes, the future volatility changes, the future mean reversion changes.

The short-rate volatility is potentially time dependent, and the continuous process of the short-term interest rate is given by:

$$d\ln(r) = \left\{\theta(t) + \frac{d\sigma(t)}{\sigma(t)} \ln(r)\right\} dt + \sigma(t) dV$$

where the factor in front of ln(r) is the **speed of mean reversion** (“gravity”), and $\theta(t)$ divided by the speed of mean reversion is a **time-dependent mean-reversion level**..

$\theta(t)$ and $\sigma(t)$ are chosen so that the model fits the term structure of spot interest rates and the term structure of spot rate volatilities. The changes in the short rate are log normally distributed, with positive interest. Once $\theta(t)$ and $\sigma(t)$ are chosen, the future short-rate volatility becomes deterministic. However, for certain specified volatility function $\sigma(t)$, the short rate can be mean-fleeing rather than mean-reverting.

The model has the advantage that the volatility unit is a percentage, conforming to the market convention. Unfortunately, due to its log normality, neither analytic solutions for the prices of bonds or the prices of bond options are available, nor numerical procedures are required to derive the short-rate tree that correctly returns the market term structures. Note that this model does not have an **affine term structure** since the volatility term is proportional to the level of the short rate. Many practitioners choose to fit the rate structure only, holding the future short-rate volatility constant.

The **convergent limit** therefore reduces to the following:

$$d\ln(r) = \theta(t)dt + \sigma(t)dV.$$

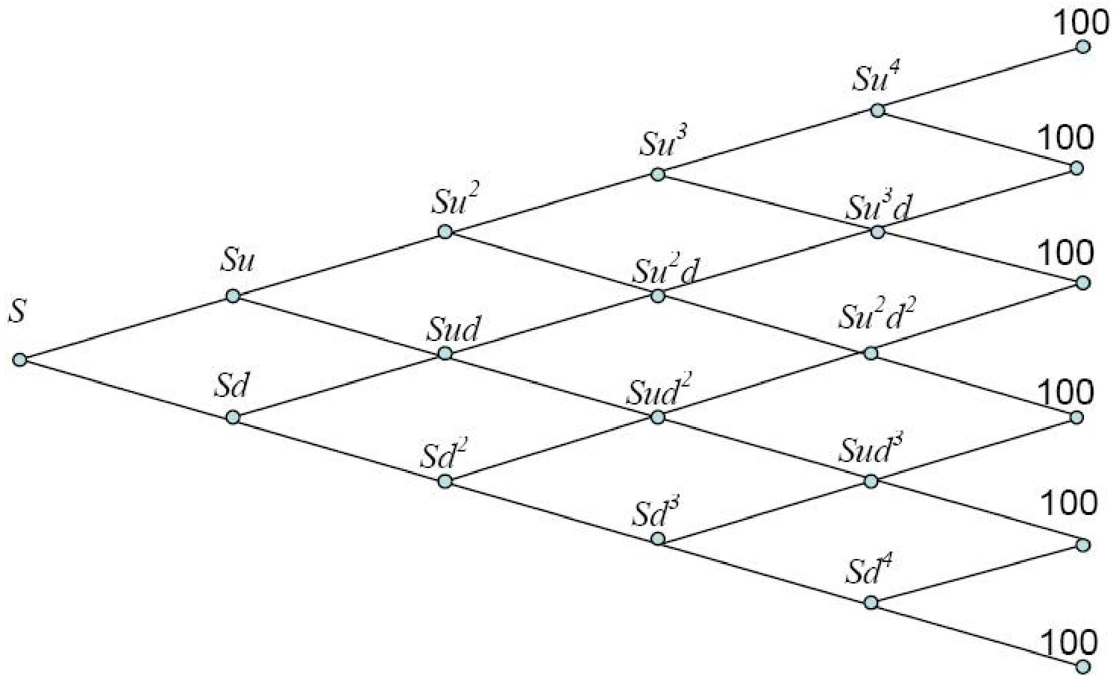
Practical Approach

Consider the value of an American call option on a five-year zero-coupon bond with time to expiration of four years and a strike price of 85.50. The term structure of zero-coupon rates and volatilities is shown in the table below. From the rates and volatilities, we will calibrate the Black-Derman-Toy interest rate tree. To price the option by using backward induction, we build a tree for the bond prices, as shown below.

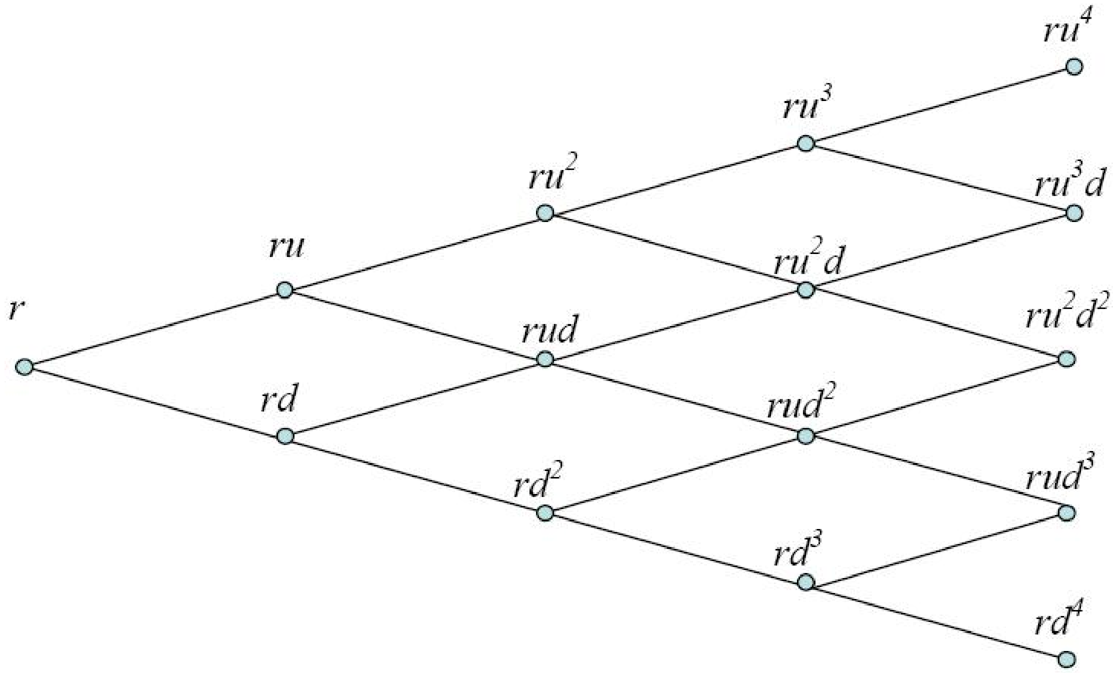
The input data:

Input to Black-Derman-Toy Model		
Years to Maturity	Zero-Coupon Rates (%)	Zero-Coupon Volatilities
1	9.0	24.0
2	9.5	22.0
3	10.0	20.0
4	10.5	18.0
5	11.0	16.0

The price tree:



The rates tree:



Valuing Securities

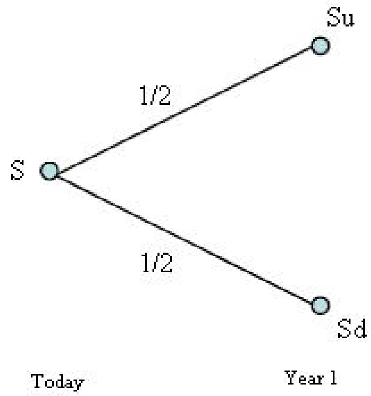
Suppose we own an interest-rate-sensitive security worth S today. We assume that its price can move up to S_u or down to S_d with equal probability over the next time period. Figure A shows the possible changes in S for a one-year time step, starting from a state where the short rate is r.

The expected price of S one year from now is ½(S_u+S_d)/S. Because we assume that all expected returns are equal, and because we can lend money at r, we deduce:

$$S = \frac{\frac{1}{2}S_u + \frac{1}{2}S_d}{1 + r}$$

Where r is today's short rate.

Figure A A One-Step Tree

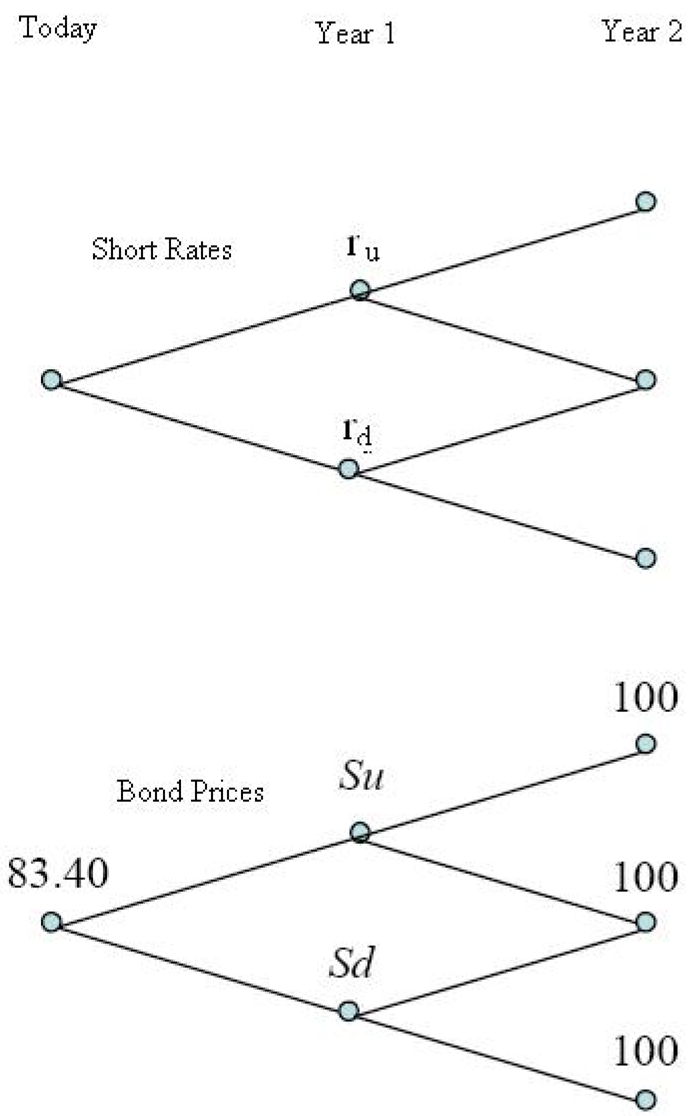


Getting Today's Prices from Future Prices

We can use the one-step tree to relate today's price to the prices one step away. Similarly, we can derive prices one step in the future from prices two steps in the future. In this way, we can relate today's prices to prices two steps away.

Figure B shows two-step trees for rates and prices.

Figure B



We find the prices of the zero-coupon bonds with maturity from one year to five year in the future. The face values are 100 (% of the nominal amount), the zero-coupon rates are given in the table above.

$$\frac{100}{1+0.09} = 91.74$$

$$\frac{100}{(1+0.095)^2} = 83.40$$

$$\frac{100}{(1+0.10)^3} = 75.13$$

$$\frac{100}{(1+0.105)^4} = 67.07$$

$$\frac{100}{(1+0.11)^5} = 59.35$$

Therefore we get the following table of data

Years to Maturity	Zero-Coupon Rates (%)	Zero-Coupon Volatilities	Zero-Bond Prices
1	9.0	24.0	91.74
2	9.5	22.0	83.40
3	10.0	20.0	75.13
4	10.5	18.0	67.07
5	11.0	16.0	59.35

Appealing to risk-neutral valuation, the following relationship must hold:

$$\frac{0.5 \times \frac{100}{1+r_d} + 0.5 \times \frac{100}{1+r_u}}{1+0.09} = 83.40$$

In a standard binomial tree, we have

$$u = e^{\sigma\sqrt{\Delta t}}, d = e^{-\sigma\sqrt{\Delta t}}$$

$$\frac{u}{d} = e^{2\sigma\sqrt{\Delta t}} \quad \rightarrow \quad \ln\left(\frac{u}{d}\right) = 2\sigma\sqrt{\Delta t}$$

$$\sigma = \frac{1}{2\sqrt{\Delta t}} \ln\left(\frac{u}{d}\right)$$

$$\begin{cases} S_u = \dfrac{100}{1+r_u} \\ S_d = \dfrac{100}{1+r_d} \end{cases}$$

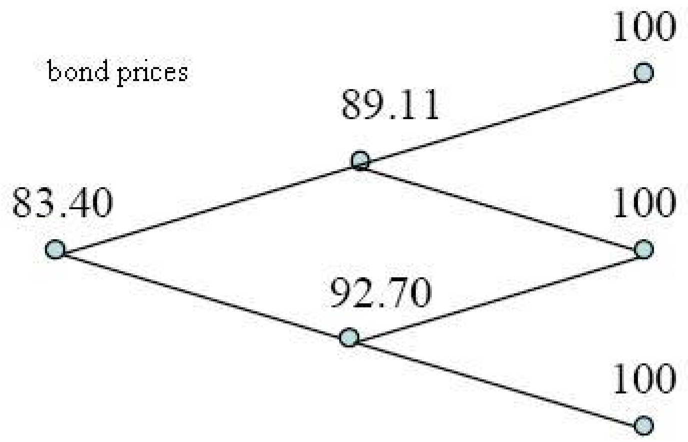
Similarly, in the Black-Derman-Toy tree, the rates are assumed to be log-normally distributed. This implies that

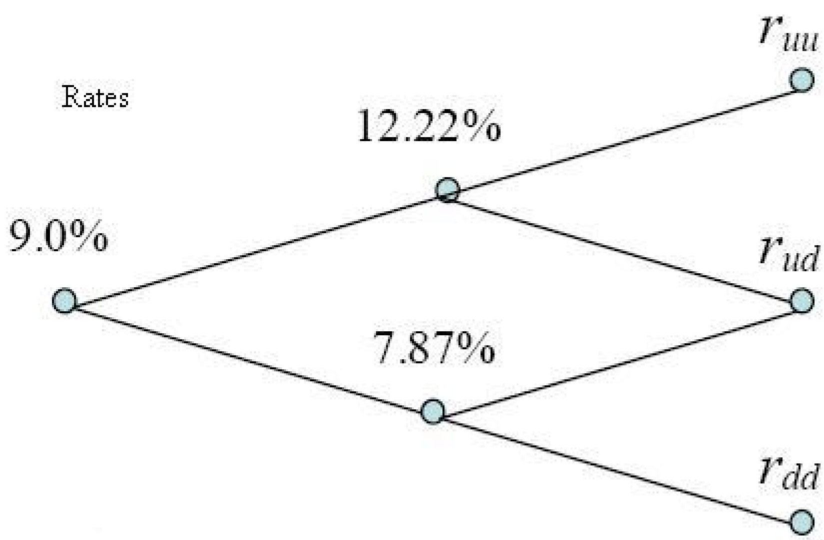
$$\sigma_n = \frac{1}{2\sqrt{\Delta t}} \ln\left(\frac{r_u}{r_d}\right)$$

Then it is very easy to find

$$r_u = 7.87\%, \qquad r_d = 12.22\%$$

So that we build the two-step tree of prices and rates:





Last time, there were two unknown rates, and two sources of information:

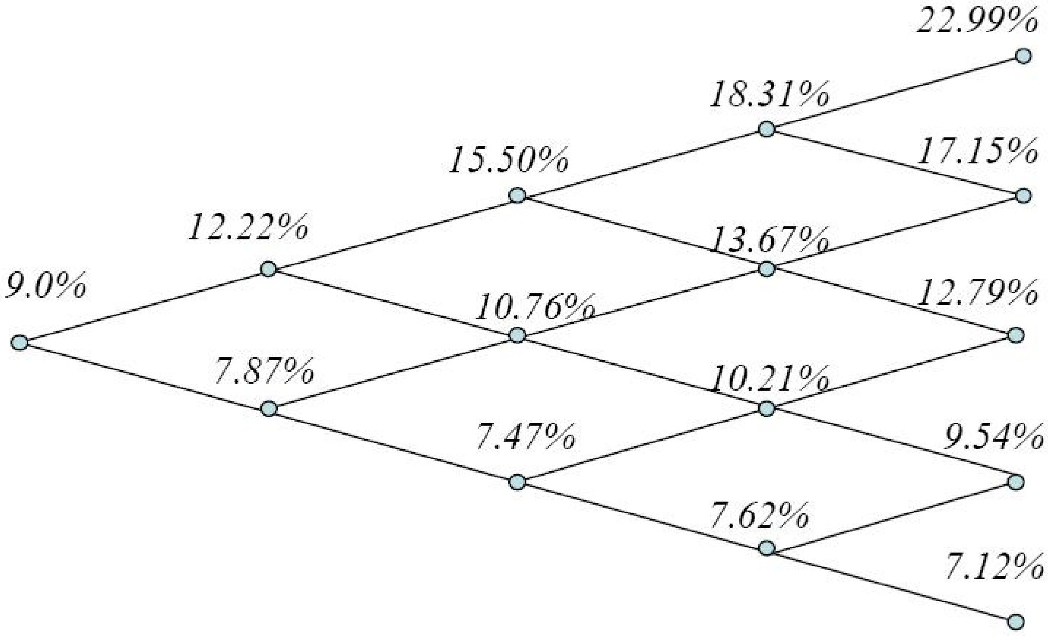
1. Zero-coupon rates.
2. The volatility of the zero-coupon rates.

This time, we have three unknown rates, but still only two sources of information. To get around this problem, remember that the Black-Derman-Toy model is built on the following assumptions:

1. Rates are lognormally distributed.
2. The volatility is only dependent on time, not on the level of the short rates. There is

thus only one level of volatility at the same time step in the rate tree.

After applying the same method above, we find the missing information in the two-period rate tree. The consecutive time steps can be computed by forward induction, as introduced by Jamshidian (1991), or more easily with the Bjerksund and Stensland (1996) analytical approximation of the short-rate interest-rate tree. Finally, we get the four-year short-rate
tree



The four-year rate tree supplies input to the solution to the five-year price tree:

					100
				81.30	
			70.44		100
		63.63		85.36	
	60.17		75.54		100
59.35		71.41		88.66	
	69.20		81.64		100
		77.89		91.29	
			85.78		100
				93.35	
					100

Excel/VBA Application

Our application of BDT-tree consists four EXCEL sheets, which will be presented later on in this section, respectively.

Sheet1: Data. We are aimed to create BDT-trees of a zero coupon bond with maturity 10.75years, time step (Delta) = 0.25, in this case, zero coupon yield and its volatility are given in the data. Thus, we start by calculating 3-month yield volatility. Note this is a real market data downloaded from internet.

Value date 2008-12-1

Maturity, T (years)	Zero Coupon Yield	Yield Volatility	Yield Volatility - 3m		B'0s
1.00	4.610%	-			0.9559303
1.25	4.511%	17.11%	8.55%		0.9463437
1.50	4.479%	16.78%	8.39%		0.9363855
1.75	4.447%	16.85%	8.42%		0.9266822
2.00	4.469%	16.46%	8.23%		0.9162722
2.25	4.527%	16.72%	8.36%		0.9051767
2.50	4.591%	17.08%	8.54%		0.8938391
2.75	4.642%	17.53%	8.76%		0.8826946
3.00	4.671%	17.37%	8.69%		0.8720147
3.25	4.705%	17.38%	8.69%		0.8612092
3.50	4.739%	17.45%	8.72%		0.8504032
3.75	4.768%	17.48%	8.74%		0.8397332
4.00	4.802%	17.10%	8.55%		0.8289335
4.25	4.841%	16.86%	8.43%		0.8179928
4.50	4.883%	16.66%	8.33%		0.8069211
4.75	4.918%	16.44%	8.22%		0.7961017
5.00	4.954%	16.18%	8.09%		0.7852401
5.25	4.994%	15.99%	8.00%		0.7742723
5.50	5.036%	15.81%	7.91%		0.7632081
5.75	5.073%	15.63%	7.81%		0.7523538
6.00	5.105%	15.37%	7.69%		0.7417606
6.25	5.139%	15.18%	7.59%		0.7310934
6.50	5.176%	14.98%	7.49%		0.7203603
6.75	5.208%	14.79%	7.40%		0.7098487
7.00	5.241%	14.60%	7.30%		0.6993665
7.25	5.276%	14.46%	7.23%		0.6888265
7.50	5.311%	14.33%	7.16%		0.6783233
7.75	5.345%	14.21%	7.10%		0.6679443

8.00	5.369%	14.01%	7.01%	0.6580908
8.25	5.395%	13.85%	6.92%	0.6482353
8.50	5.422%	13.68%	6.84%	0.6383856
8.75	5.447%	13.52%	6.76%	0.6287091
9.00	5.471%	13.34%	6.67%	0.6191378
9.25	5.498%	13.21%	6.60%	0.6095310
9.50	5.526%	13.07%	6.54%	0.5998931
9.75	5.551%	12.94%	6.47%	0.5905103
10.00	5.577%	12.79%	6.40%	0.5811906
10.25	5.604%	12.69%	6.34%	0.5718434
10.50	5.633%	12.57%	6.28%	0.5624729
10.75	5.659%	12.46%	6.23%	0.5533492

Table1 Input Data with Corresponding Zero Coupon Bond Prices

Sheet2: Yield Curve. To examine the relationship between interest rate and time to maturity in each step, we therefore made a yield cure.

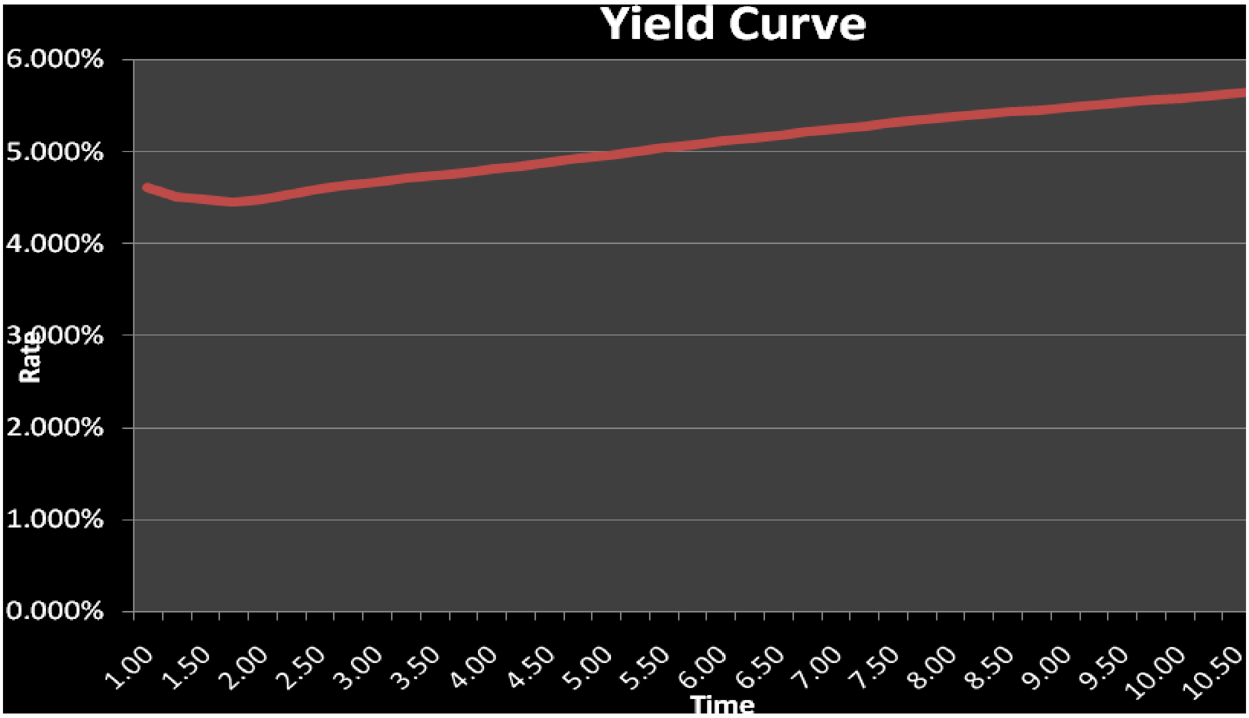


Figure1 The Yield Curve

Sheet3: Rate Tree VBA. It can be considered as the major part of our application, since in this part the BDT matrix is calculated by user-defined functions within VBA. In this case, we define the following function (the processes of programming each function will be shown in APPENDIX):

- ✧ <u>Function Bzero</u> - calculates zero coupon bond yields based on continuously compounded interest rates.
- ✧ <u>Function BDTtree</u> - determines the rate tree according to BDT model, which

depends upon the parameters volatility, face value, delta as well as bond prices represented by the previous function Bzero on a continuous basis. Newton-Raphson method has been used during the calculation in order to optimize the algorithm.

- ✧ Function Bond – calculates bond prices by the inputs face value, interest rate, periods, and delta (time step).

	0.5	0.75	1	1.25	1.5	1.75	2
19.76%	4.47%	5.08%	5.42%	6.35%	7.47%	8.47%	9.28%
0.00%	3.76%	4.29%	4.58%	5.38%	6.32%	7.14%	7.79%
0.00%	0.00%	3.63%	3.87%	4.57%	5.35%	6.02%	6.54%
0.00%	0.00%	0.00%	3.27%	3.87%	4.52%	5.07%	5.49%
0.00%	0.00%	0.00%	0.00%	3.28%	3.83%	4.28%	4.61%
0.00%	0.00%	0.00%	0.00%	0.00%	3.24%	3.60%	3.87%
0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	3.04%	3.24%
0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	2.72%
0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Figure2 Part of BDT Matrix

Sheet4: Rate Tree. Due to the large size of our data, here we only show part of the obtained rate tree instead of complete one. Part of the obtained BDT rate-tree from time 0 to 2 with step 0.25 is shown in the following table.

Time, t	0.25	0.5	0.75	1	1.25	1.5	1.75	2
Volatility, σ		8.55%	8.39%	8.42%	8.23%	8.36%	8.54%	8.76%
K		1.186598	1.182688	1.183489	1.178953	1.181971	1.186287	1.191552
								0.090102
							0.081501	
						0.072051		0.075617
					0.062558		0.068702	
				0.055468		0.060958		0.063461
			0.05189		0.053062		0.057914	
		0.047888		0.046868		0.051573		0.053259
	0.046101		0.043875		0.045008		0.048819	
		0.040357		0.039602		0.043633		0.044697
			0.037097		0.038176		0.041153	
				0.033462		0.036916		0.037512
					0.032381		0.034691	
						0.031232		0.031482
							0.029243	
								0.026421

Sheet5: Bond Tree. Ultimately we computed bond prices assocated with each node in the above rate tree, the results are presented in this worksheet. Similarly the prices tree in fixed horizon will be shown below.

Time, t	0.25	0.5	0.75	1	1.25	1.5	1.75	2
								46.44788
							47.94466	
						49.50421		51.33819
					51.12644		52.80091	
				52.78166		54.31165		56.03246
			54.42613		55.87123		57.44121	
		56.06467		57.45602		58.88453		60.47833
	57.6673		59.02216		60.36419		61.81776	
		60.57682		61.86231		63.17995		64.63929
			63.33576		64.56758		65.89851	
				65.96801		67.17042		68.49325
					68.45857		69.66535	
						70.8419		72.03042
							73.112	
								75.25107

Table2 BDT Bond Price Tree

Reference

1. Jan R.M. Röman, Lecture notes in Analytical Finance II, August 15, 2007.
2. Mary Jackson and Mike Staunton, Advanced modeling in finance using Excel and VBA, John Wiley & Sons, LTD
3. John C. Hull, Options, Futures, and Other Derivatives. - Prentice Hall, 2006, sixth edition. - ISBN 0-13-149908-4
4. http://en.wikipedia.org/wiki/Black-Derman-Toy, last visit Dec.4, 2008
5. Black, F.; Derman, E. and Toy, W. (January-February 1990). "A One-Factor Model of Interest Rates and Its Application to Treasury Bond Options". *Financial Analysts Journal*: 24–32,

Suggestions for Further Reading

1. T.G., Bali, Karagozoglu “Implementation of the BDT-Model with Different Volatility Estimators”, Journal of Fixed Income, March 1999, pp.24-34
2. Fischer Black, Piotr Karasinsky , “Bond and Option Pricing when Short Rates are Lognormal”, Financial Analysts Journal, July-August 1991, pp.52-59
3. John Hull, Alan White, “New Ways with the Yield Curve”, Risk, October 1990a
4. John Hull, Alan White, “One Factor Interest-Rate Models and the Valuation of Interest-Rate Derivative Securities”, Journal of Financial and Quantitative Analysis, Vol. 28, June 1993, pp.235-254
5. John Hull, Alan White, “Pricing Interest-Rate Derivative Securities”, Review of Financial Studies, Vol. 4, 1990b, pp.573-591
6. R. Jarrow and S. Turnbull “Pricing Derivatives on Financial Securities subject to Credit Risk”. The Journal of Finance, March 1995, pp. 53-85
7. R. Jarrow, D. Lando and Turnbull, “A Markov Model for the Term Structure of Credit Risk Spreads”. The review of financial studies, 1997 Vol. 10, No.2.

Appendix

Finally we would like to enclose VBA code associated with user-defined function in EXCEL. Commends (in green color) have been added where necessary.

```
Function BZero(Contdis, Face, rate, Maturity, delta)

'This function calculates the B0's and has continuous rates as parameters:
'Contdis: 1 stands for continuous discount factor while 2 for discrete
'Face: value of the obligation at maturity
'Rate: interest rate of BDT tree
'Maturity: maturity of the bond
'Delta: interval of time to adjusted rate
'Definition of variables
Dim p, pstar
'p is the probability of up and pstar is a probability of down
Dim i As Integer, j As Integer
Dim cfvec() As Variant
'cfvec is a vector of the bonds' cash-flows
'i is an element of cfvec and j is the step
ReDim cfvec(Maturity + 1)
'At starting point 1 the matrix has only one dimension
'Definition of probabilities under BDT
p = 0.5
pstar = 1 - p
'The final cash flow is equal to Face
For i = 1 To Maturity + 1
    cfvec(i) = Face
Next i
'Update of cash flows required in terms of continuous
'From maturity back to 1 with steps -1
'Restored value of i element of the vector is equal to 0.5*i element "old" (up)
'0.5 * element "old" (down) updates on a continuous basis at the corresponding rate
'BDT tree adjusted to the time
For j = Maturity To 1 Step -1
    For i = 1 To j
        If Contdis = 1 Then cfvec(i) = (p * cfvec(i) + pstar * cfvec(i + 1)) * Exp(-rate(i, j) * delta)
        If Contdis = 2 Then cfvec(i) = (p * cfvec(i) + pstar * cfvec(i + 1)) / ((1 + rate(i, j)) ^ delta)
    Next i
Next j
'Bzero is the first element of final value of cfvec
```

```
BZero = cfvec(1)
End Function
```

Function BDTTree(Contdis, Bzeros, Volatility, Face, delta)

```
'This function determines the rate tree according to BDT model
'Contdis: 1 for continuous and 2 for discrete
'Bzeros: zero coupon bond yield
'Volatility: vlatility of yields
'Face: nominal value of the bond at maturity
'Delta: time interval to adjusted rate
'Use the above BZero
'It uses Newton-Raphson method as algorithm optimization
'The unknown is the maximum rate which determines the other on the basis of this condition and volatility
'Definition of variables
Dim movement, tolerance, bzerok, volk, rmaxk, fr1, f, derivative
'movement: change in the maximum rate
'tolerance: value of the maximum allowable error iterations
'bzerok: Bzero corresponding to the element of vector k
'volk: volatility of the yield corresponding to bzerok
'rmaxk: Maximum Rate
'FR1: function resulting from a small change (movement) of the rate
'f: original function
'derivative: derivative of f
Dim i As Integer, j As Integer, k As Integer, Maturity As Integer
Dim RateMatrix() As Variant
'RateMatrix is a matrix reflects the changes in BDT tree
'The function counts the number of elements of Bzeros - spot prices.
Maturity = Application.Count(Bzeros)
'Matrix dimension depends on the counted value
ReDim RateMatrix(Maturity, Maturity)
'Specification of the differences in the rate and tolerance for stopping procedure
movement = 0.0001
tolerance = 0.00000001
'Begin from the first rate of the matrix
If Contdis = 1 Then RateMatrix(1, 1) = -Log(Bzeros(1) / Face) / delta
If Contdis = 2 Then RateMatrix(1, 1) = (Face / Bzeros(1)) ^ (1 / delta) - 1
'Determine the rate column by column, starting at 2 because the first value is known
For k = 2 To Maturity
'Identify price of B0 observed for in period of calculation k
bzerok = Bzeros(k)
'Provide the volatility of the corresponding period (the vector has less volatility)
'Vector of prices adjusted for the period
```

```
volk = 2 * Volatility(k - 1) * Sqr(delta)
'Use Newton-Raphson as algorithm optimization to solve the maximum rate (rmaxk)
'Current maximum rate is determined by the maximum rate found in previous period
rmaxk = RateMatrix(1, k - 1)
Do
'Estimate function due to changes in maximum rate
RateMatrix(1, k) = rmaxk + movement
For i = 2 To k
    RateMatrix(i, k) = RateMatrix(i - 1, k) * Exp(-volk)
Next i
     fr1 = BZero(Contdis, Face, RateMatrix, k, delta) - bzerok
'Estimate original function based on the maximum rate
RateMatrix(1, k) = rmaxk
For i = 2 To k
    RateMatrix(i, k) = RateMatrix(i - 1, k) * Exp(-volk)
Next i
    f = BZero(Contdis, Face, RateMatrix, k, delta) - bzerok
'f is equal to the difference between the calculated and observed values
'Cancel the iterations
derivative = (fr1 - f) / movement
'Estimation with Newton's equation:
rmaxk = rmaxk - (f / derivative)
'Iterations of the original function is close to zero.
Loop While Abs(f) > tolerance
Next k
BDTTree = RateMatrix
'BDTTree gets the final value of Rate Matrix
End Function
```

Function BOND(Face, rate, periods, delta)

```
'Definition of varibles
Dim p, pstar
Dim i As Integer, j As Integer
Dim bondvec() As Variant
'Redefine the vectors
ReDim bondvec(periods + 1, periods + 1)
'Definition of probabilities
p = 0.5
pstar = 1 - p
'Value at maturity
For j = periods + 1 To periods + 1
For i = 1 To periods + 1
     bondvec(i, periods + 1) = Face
```

```
Next i
Next j
'Calculate the results for current time period
For j = periods To 1 Step -1
For i = 1 To j
bondvec(i, j) = (p * bondvec(i, j + 1) + pstar * bondvec(i + 1, j + 1)) / ((1 + rate(i, j)) ^
delta)
Next i
Next j
'Solution is a vector
BOND = bondvec()
End Function
```