Implementation of Arbitrage-free Discretization of Interest Rate Dynamics and Calibration via Swaptions and Caps in Excel VBA.

Ohoe Kim, Swathi Gaddam

Department of Mathematics, Towson University, Towson, Maryland 21252
(Email: okim@towson.edu, swathi1982@gmail.com)

Abstract. We consider Libor market model and calibration process. We estimate some interest rate derivatives such as Caps and Swaptions. We use Monte Carlo methods with appropriate dynamics under spot measure and forward measure. Imposing martingale discretization property, the same quantities are estimated again to see a contrast or discrepancy in light of confidence interval. Based on algorithms discussed in the paper, "Arbitrage-free discretizations of lognormal forward Libor and swap rate models" by Glasserman, P and Zhao, X (2000), the algorithms are implemented in excel VBA. Starting with current market data such as FRAs and Bond Prices, we describe steps involved to estimate the price of interest rate derivatives such as Caps and Swaptions, which would be beneficial to the practitioners in the industry.

Key words: Interest Rate Models, Martingale, Monte Carlo Simulation, Libor Market Models, Excel VBA

AMS Classification (2000): 60G42, 60G44, 60G15, 60J60, 65C05

1. *Introduction*

The *LIBOR Market Model* (LMM) has become "the industry standard model" for pricing interest rate derivatives. Based on the Heath-Jarrow-Morton (HJM) forward rate approach, it builds a process for LIBOR interest rates, assuming a conditional lognormal process for LIBOR. In practice, market interest rates are based on simple compounding over time intervals. The discretized model is used for pricing. These features of continuous-time formulations are lost when the models are discretized for simulation.

Therefore, we demonstrate the methods for discretizing the most standard *interest rates model*, LMM London Inter Bank Offered Rates Market model by preserving main principles of a market model, under risk neutral measure and forward measure. LIBOR rates are modeled based on observable market rates and are based on simple interest. Calibrating this model depends on choosing *volatility* to match market price.

We shall study the impact and structure of LMM under different term structure of volatilities in time. Also, we study their impact on the market and on calibration of interest derivatives such as the Caps and Swaptions markets.

We utilize the algorithms discussed in the paper of Glasserman, P and Zhao, X (2000). We implement these algorithms starting with market data of bond price or FRAs under spot measure and forward measure to generate LIBOR path, calibrated by the market volatility implied by Caps and Swaptions.

Our aim is to benefit market practitioners for them to allow them to generate path of LIBOR and estimate the prices of interest rate derivatives such as Caps and Swaptions.

2. *Preliminaries*

Our goal to discretize the dynamics of Forward LIBOR rate model that is described by a system of stochastic differential equations of the form

$$\frac{dL_n(t)}{L_n(t)} = \mu_n(t)dt + \sigma_n(t)dW(t) \tag{1}$$

Where

- $0 \le t \le T_n, n = 1,\dots,m$
- $W - d$ -Dimensional Standard Brownian Motion
- μ_n, σ_n depend on the current
- μ_n, σ_n depends on vector $< L_1(t),\dots,L_m(t) >$ as well as the time t
- $L_n(t)$ is the Forward Libor Rate fixed at time t for the interval $[t_n, t_{n+1}]$

2.1. LMM Under Spot Measure.

When the numeraire asset is money market fund,

$$\beta^*(t) = B_{y(t)}(t) \prod_{j=0}^{y(t)-1} [1 + \delta_j L_j(T_j)] \tag{2}$$

where $y(t)$ gives the index of the next tenor date at time t, then the associated measure is called the *spot measure*. $B_n(t)$ is the bond price at time t maturing at t_n.

Deflated bond price, $D_n(t) = \dfrac{B_n(t)}{\beta^*(t)}$ is *martingale* with respect to spot measure, and

$$D_n(t) = \frac{B_n(t)}{\beta^*(t)} = \prod_{j=y(t)}^{n-1} \frac{1}{1+\delta_j L_j(t)} \prod_{j=0}^{y(t)-1} \frac{1}{1+\delta_j L_j(T_j)} \tag{3}$$

If the D_n is a positive *martingale* then $\dfrac{dD_{n+1}(t)}{D_{n+1}(t)} = V_{n+1}^T(t)dW(t)$ where $n = 1,\dots,m$

By Ito's Lemma,

$$dLogD_{n+1}(t) = -\frac{1}{2}\left\| V_{n+1}(t) \right\| dt + V_{n+1}^{T} dW(t) \qquad (4)$$

For one Factor Model, $\frac{dD_{n+1}(t)}{D_{n+1}(t)} = V_{n+1}(t)dW(t) \Rightarrow dLogD_{n+1}(t) = -\frac{1}{2}V_{n+1}(t)dt + V_{n+1}(t)dW(t)$

$$dLogD_{n+1}(t) = -\sum_{j=y(t)}^{n} dLog(1+\delta_j L_j(t)) \qquad (5)$$

Use Ito's Lemma and (L.1), we get

$$V_{n+1}(t) = -\sum_{j=y(t)}^{n} \frac{\delta_j L_j(t)}{1+\delta_j L_j(t)} \sigma_j(t) \qquad (6)$$

Above shows the relation between volatility of bond and LIBOR

From which we claim,

$$\mu_n = -\sigma_n V_{n+1} = \sum_{j=y(t)}^{n} \frac{\delta_j L_j(t)}{1+\delta_j L_j(t)} \sigma_j(t)\ \sigma_n(t) \qquad (7)$$

Now (L.1) changes to

$$\frac{dL_n(t)}{L_n(t)} = \mu_n(t)dt + \sigma_n^T(t)dW(t) = \sum_{j=y(t)}^{n} \frac{\delta_j L_j(t)}{1+\delta_j L_j(t)} \sigma_n^T(t)\sigma_j(t) + \sigma_n^T(t)dW(t) \qquad (8)$$

where $0 \le t \le T_n, n = 1,...,m$. This describes arbitrage free dynamics and forward LIBOR rates under the spot measure. Similar to HJM approach, drift is determined once the volatility is specified.

2.2. LMM Under Forward Measure.

Derivative asset pricing is based on converting prices of such assets into martingales. LIBOR rates are not martingale under the spot measure. There is no discrete drift that preserves martingale property under spot measure. This motivates for the next approach which is by using forward measure. In light of Girsanov theorem, we can change the underlying probability measure to forward measure which in turn changes the dynamics. The dynamics, expressed under the new measure, has no drift term. Then using Monte Carlo simulation with respect to the new measure, we can simulate the arbitrage free pricing of interest rate derivatives such as swaption and caps.

When numeraire asset is bond maturity at T_{m+1}, B_{m+1}, then the associated measure is called f*orward measure*. Deflated bond price, $D_n(t) = \dfrac{B_n(t)}{B_{m+1}(t)} = \prod_{j=n}^{m}(1+\delta_j L_j(t))$ is martingale with respect to forward measure, P_{m+1}. If D_n is a positive martingale, then

$$\frac{dD_{n+1}(t)}{D_{n+1}(t)} = V_{n+1}^T(t)dW(t) \quad \text{where } n = 1,\ldots,m$$

The arbitrage free dynamics of the L_n under P_{m+1}:

$$\frac{dL_n(t)}{L_n(t)} = -\sum_{j=n+1}^{m} \frac{1}{1+\delta_j L_j(t)} \delta_j L_j(t)\sigma_n^T(t)\sigma_j(t)dt + \sigma_n^T(t)dW^{m+1}(t)$$

When $n = m$, $\dfrac{dL_m(t)}{L_m(t)} = \sigma_m^T(t)dW^{m+1}(t)$.

When $n = m$, L_m is martingale with respect to P_{m+1}.

2.3. Martingale Discretization.

We use hats to distinguish discretized variables from their exact continuous time counterparts.

As we mentioned earlier, under the spot measure, the LIBOR rates are not martingale. Also, we consider special case under spot measure. When $n = 2$, $\dfrac{1}{1+\delta_1 \hat{L}_1}$ be martingale since $\dfrac{1}{1+\delta_0 \hat{L}_0}$ is constant, $\hat{D}_n(t) = \dfrac{\hat{B}_n(t)}{\beta^*(t)}$ is martingale with respect to spot measure, and

$\hat{D}_n(t) = \frac{\hat{B}_n(t)}{\beta^*(t)} = \prod_{j=y(t)}^{n-1} \frac{1}{1+\delta_j \hat{L}_j(t)} \prod_{j=0}^{y(t)-1} \frac{1}{1+\delta_j \hat{L}_j(T_j)}$. $E\left[\frac{1}{1+x}\right]$ is infinite If $x \sim N(a,b)$ for any a . So, $\frac{1}{1+\delta_1 \hat{L}_1(0)}$ is infinite no matter how we choose μ_1. There is no discrete drift that preserves martingale property under spot measure.

Even though LIBOR rates are martingale with respect to forward measure, after discretization, it looses its martingale property. Therefore, LIBOR rates are not martingale under forward measure. We need a different technique to discretize the LIBOR rates

Now, we consider martingale discretization. As in HJM setting, we derived discrete drift from the condition that discretized discounted bond price must be martingale. In the LIBOR market, the corresponding requirement is:

- **Under the spot measure**, $\hat{D}_n(t_i) = \prod_{j=0}^{n-1} \frac{1}{1+\delta_j \hat{L}_j(t_i)}$ be martingale in every i for each n
- **Under the forward measure**, the martingale condition is $\hat{D}_n(t_i) = \prod_{j=n}^{m} 1+\delta_j \hat{L}_j(t_i)$

Therefore, we simulate the deflated bond price themselves rather than the forward LIBOR rates. Also, under the spot measure, the deflated bond price satisfy,

$$\frac{dD_{n+1}(t)}{D_{n+1}(t)} = -\sum_{j=y(t)}^{n} \frac{\delta_j L_j(t)}{1+\delta_j L_j(t)} \sigma_j^T(t) dW(t) = \sum_{j=y(t)}^{n} \left(\frac{D_{j+1}(t)}{D_j(t)} - 1 \right) \sigma_j^T(t) dW(t)$$

Under forward measure, the deflated bond price satisfy,

$$\frac{dD_{n+1}(t)}{D_{n+1}(t)} = \sum_{j=n+1}^{m} \frac{\delta_j L_j(t)}{1+\delta_j L_j(t)} \sigma_j^T(t) dW^{m+1}(t) = \sum_{j=n+1}^{m} \left(\frac{1-D_{j+1}(t)}{D_j(t)} - 1 \right) \sigma_j^T(t) d^{m+1}W(t)$$

3. Implementation of algorithms using Excel VBA

3.1. Discretization of LMM Under Spot Measure for Simulation Purpose.

The application of Euler scheme for $\log L_n$ under spot measure discretizes the SDE producing

$$\hat{L}_n(t_{i+1}) = \hat{L}_n(t_i)\exp[\{\mu_n(\hat{L}_n, t_i)\, t_i - \frac{1}{2}\|\sigma_n(t_i)\|^2\}(t_{i+1}-t_i) + \sqrt{t_{i+1}-t_i}\,\sigma_n{}^T(t_i)Z_{i+1}]$$

$$\text{with } \mu_n(\hat{L}(t_i), t_i) = \sum_{j=y(t_i)}^{n} \frac{\delta_j \hat{L}_j(t_i)\sigma_n{}^T(t_i)\sigma_j(t_i)}{1+\delta_j \hat{L}_j(t_i)}$$

and Z_i 's are independent $N(0,1)$. Drift of LIBOR Rates, under spot measure, are controlled in such a way that the prices of interest sensitive derivatives are arbitrage free.

3.1.1. Calibration of LIBOR Under Spot Measure

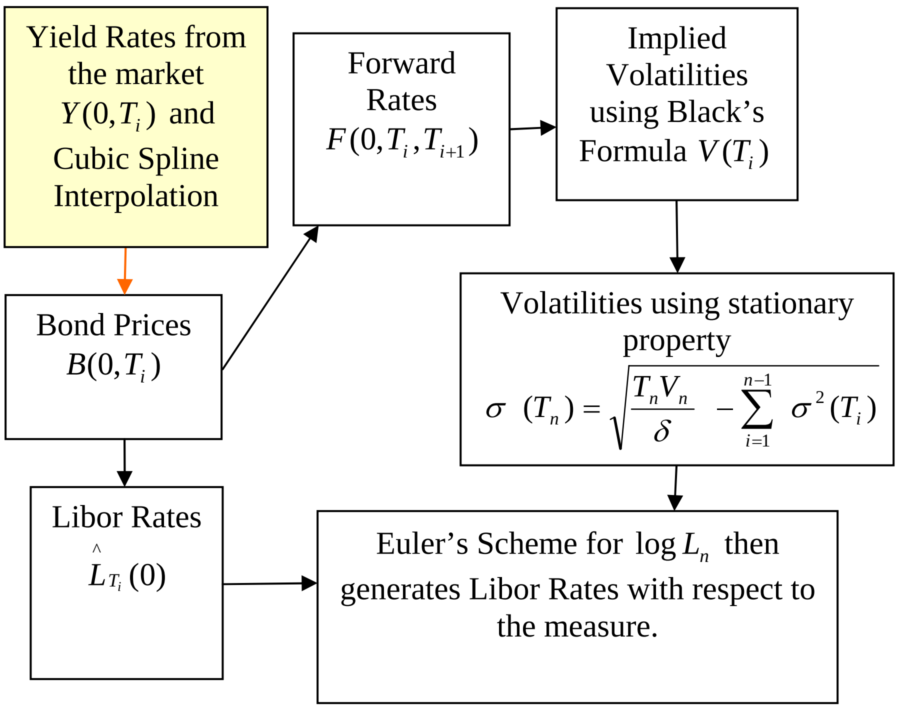
We consider following steps to generate LIBOR rates that matches term structure and then to calculate the price of Cap and Swaption

- First we can get price of the Caplet from the market, and use the Black's formula of the Caplet to obtain the implied volatility.
- Then we obtain the volatility of LIBOR by assuming stationary property
- Once we have volatility of LIBOR, we are able to generate the drift component and LIBOR rates
- Generating different paths of LIBOR, we are able to simulate arbitrage free prices of derivatives such as Caps and Swaptions
- If $\sigma_n(t) = \sigma(t, T_n)$ depends on n and t only through the difference $T_n - t$ then volatility structure is stationary.

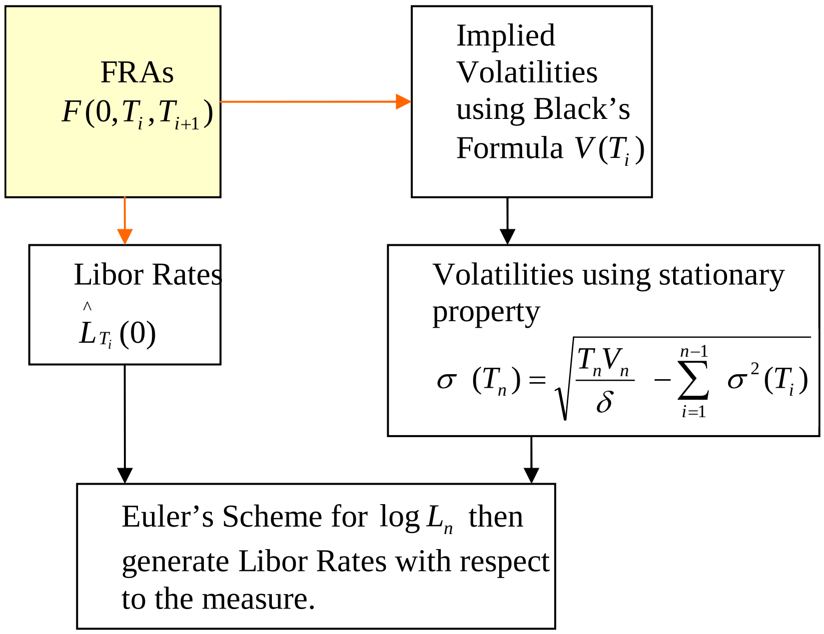
- Consider a stationary, single-factor, piecewise constant volatility structure. $\sigma(i)$ is defined as the volatility of a forward rate i periods away maturity.
- Implied volatilities $V(t_i)$ is defined as volatility of a forward rate between 0 to t_i.

Above steps are illustrated in the following flow charts.

Flow Chart 1: One Method for Calibration of LMM and Starting With Yield Rates



Flow Chart 2: Second Method for Calibration of LMM and Starting With FRAs



3.2. Discretization of LMM Under Forward Measure for Simulation Purpose.

Similarly, we use hats to distinguish discretized variables from their exact continuous time counterparts. The application of Euler scheme for $\log L_n$ under forward measure discretizes the SDE producing

$$\hat{L}_n(t_{i+1})=\hat{L}_n(t_i)\exp[\{\mu_n(\hat{L}_n,t_i)t_i-\frac{1}{2}\|\sigma_n(t_i)\|^2\}(t_{i+1}-t_i)+\sqrt{t_{i+1}-t_i}\sigma_n{}^T(t_i)Z_{i+1}]$$

$$\text{with}\ \ \mu_n(\hat{L}(t_i),t_i)=-\sum_{j=n+1}^{m}\frac{\delta_j\hat{L}_j(t_i)\sigma_n{}^T(t_i)\ \sigma_j(t_i)}{1+\delta_j\hat{L}_j(t_i)}$$

It follows that if σ_M is deterministic and constant between the tenor dates then the log Euler scheme with $\mu_M\equiv 0$ simulates L_M without discretization error under the forward measure P_{m+1}.

3.2.1. Calibration of LIBOR Under Forward Measure

Similar to what we had under spot measure, we consider following steps to generate LIBOR rates that matches term structure and then calculate the price of Cap and Swaption

- First we can get price of the Caplet from the market, and use the Black's formula of the Caplet to obtain the implied volatility.
- Then we obtain the volatility of LIBOR by assuming stationary property
- Once we have volatility of LIBOR, we are able to generate the drift component and LIBOR rates
- By generating different paths of LIBOR, we are able to simulate arbitrage free prices of derivatives such as Caps and Swaptions

As we did under spot measure, we consider the same Flow Charts for Calibration of LMM one starting with yield rates and other starting with FRA’s, under forward measure.

3.3. Martingale Discretization.

3.3.1. Under the Spot Measure

Under the spot measure, the deflated bond price satisfy,

$$\frac{dD_{n+1}(t)}{D_{n+1}(t)} = -\sum_{j=y(t)}^{n} \frac{\delta_j L_j(t)}{1+\delta_j L_j(t)} \sigma_j^T(t) dW(t) = \sum_{j=y(t)}^{n} \left(\frac{D_{j+1}(t)}{D_j(t)} - 1\right) \sigma_j^T(t) dW(t)$$

Using Euler’s Scheme for $LogD_{n+1}$,

$$\hat{D}_{n+1}(t_{i+1}) = \hat{D}_n(t_i) \exp[-\frac{1}{2} \| \hat{V}_{n+1}(t_i) \|^2 (t_{i+1} - t_i) + \sqrt{t_{i+1} - t_i}\, \hat{V}_{n+1}^T(t_i)\ Z_{i+1}$$

$$\text{with } \hat{V}_{n+1}(t_i) = \sum_{j=y(t_i)}^{n} \left(\frac{\hat{D}_{j+1}(t_i)}{\hat{D}_j(t_i)} - 1\right) \sigma_j(t_i)$$

Once we get arbitrage free discretized $\hat{D}_n(t_i)$ with martingale property, we can generate the forward LIBOR rates using following relation:

$$\hat{L}_n(t_i) = \frac{1}{\delta_n} \frac{\hat{D}_n(t_i) - \hat{D}_{n+1}(t_i)}{\hat{D}_{n+1}(t_i)} \text{ for } n = 1, \ldots, m$$

3.3.2. Under Forward Measure

Under the forward measure, P_{m+1} the deflated bond price must satisfy,

$$\frac{dD_{n+1}(t)}{D_{n+1}(t)} = -\sum_{j=n+1}^{m} \frac{\delta_j L_j(t)}{1+\delta_j L_j(t)} \sigma_j^T(t) dW^{m+1}(t) = \sum_{j=n+1}^{m} \left(\frac{1-D_{j+1}(t)}{D_j(t)} - 1 \right) \sigma_j^T(t) d^{m+1} W(t)$$

$$\text{Where } \hat{V}_{n+1}(t_i) = \sum_{j=n+1}^{m} \left(\frac{1-D_{j+1}(t)}{D_j(t)} \right) \sigma_j(t)$$

We refer to the paper by Glasserman and Zhao for complete proofs for discretization under martingale discretization.

3.3.3. Calibration of LIBOR Using Martingale Discretization

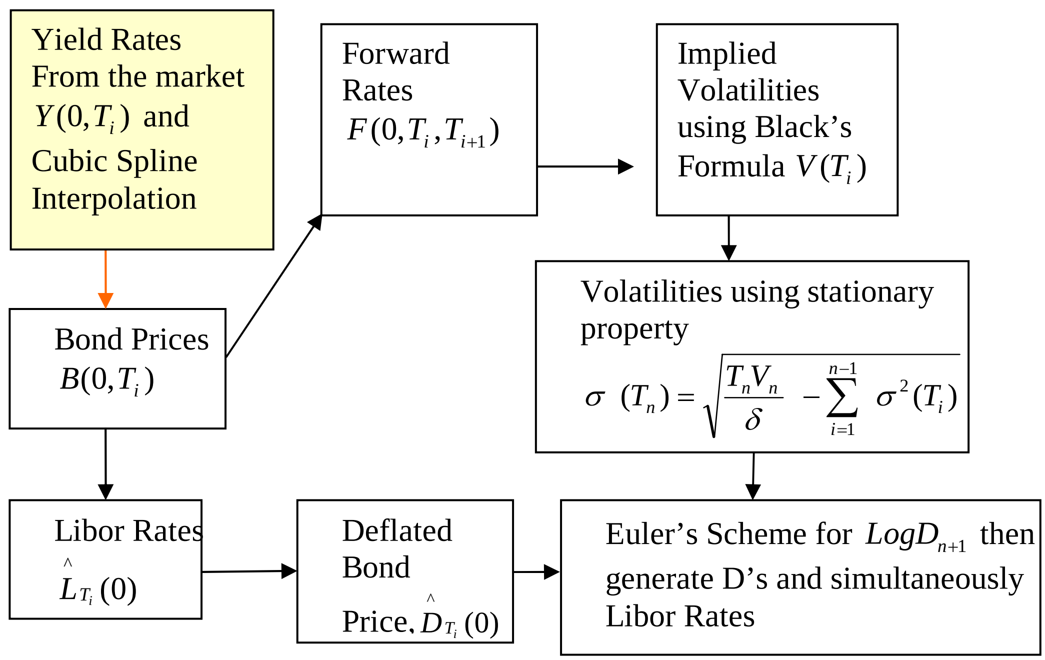
We consider following steps to generate LIBOR rates that matches term structure, to calculate the price of Cap and Swaption

- First we get the market data for yield prices
- Then we can estimate bond prices, Libor rates, and deflated bond prices at time t=0.
- Similar to what we had under spot measure where we generate LIBOR rates that matches term structure. Then calculate the price of Cap and Swaption
- We can get price of the Caplet from the market and use the Black's formula of the Caplet to obtain the implied volatility.
- Then we obtain the volatility of LIBOR by assuming stationary property
- Once we have volatility of LIBOR, we are able to generate the drift component of deflated bond prices, deflated bond prices and LIBOR rates simultaneously.

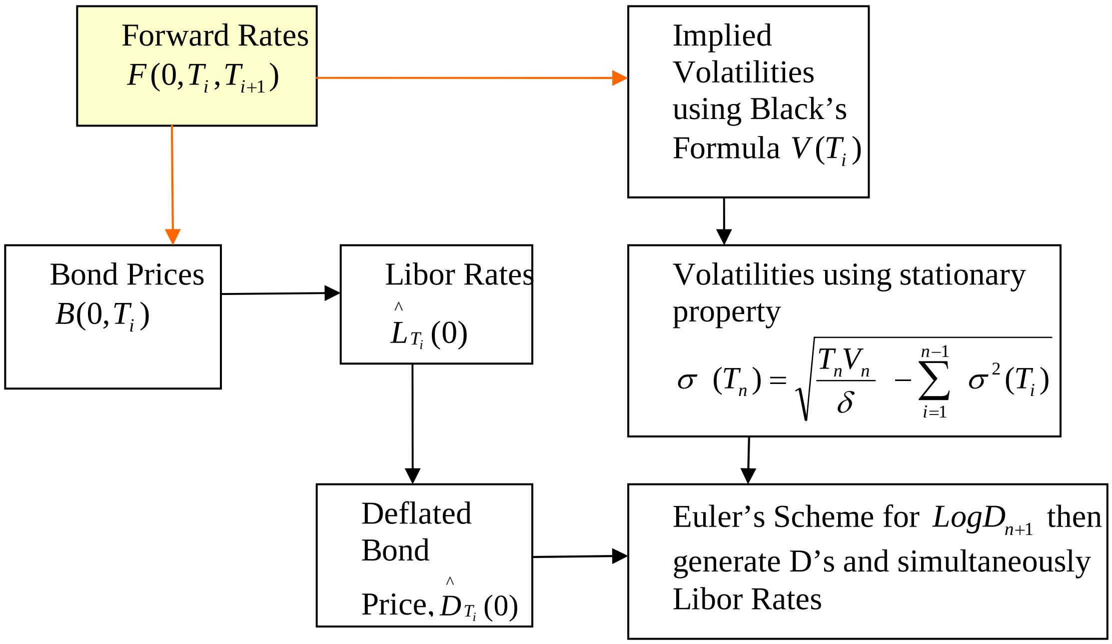
- By generating different paths of LIBOR, we are able to simulate arbitrage free prices of derivatives such as Caps and Swaptions

Above steps are illustrated in the following flow carts.

Flow Chart 3: Calibration of LMM Starting With Yield Rates



Flow Cart 4: Calibration of LMM After Martingale Discretization and Starting With FRAs



4. Numerical Experiment

Given market data (yield rates and forward rates), we generate LIBOR rates under spot and forward measure, calibrate, and then estimate the prices of Caps and Swaptions. Also, imposing martingale discretization property, the same quantities are estimated again to see a contrast or discrepancy. In this numerical experiment, we only consider a single factor model of equation (1).

This numerical experiment allows the market practitioners in the industry to generate paths of LIBOR rates and to estimate prices of interest rate derivatives such as Caps and Swaptions. Furthermore, one may make use of excel VBA code that is given in appendix for reference.

Given following data for the Yield Rates in the market.

Table 1: Yield Rates

Time	Yield Rates
0	0.04
0.25	0.037
0.5	0.04
2	0.0433
3	0.0437
5	0.0441
10	0.0455
30	0.0467

We perform cubic spline interpolation on yield rates by using tools provided by software packages such as Mathematica and Matlab, and then use the formula,

$f(0,t) = Y(0,t) + tY'(0,t)$ or use relation between forward rates and bond, to find the cubic polynomial expression for **Forward Rates** from which we can get following data:

Table 2: Forward Rates

Time (t)	$F(t)$
0	0.04
0.25	0.03716
0.5	0.04788
0.75	0.04968
1	0.04809
1.25	0.04471
1.5	0.04112
1.75	0.03891
2	0.03968
2.25	0.04252

Also, we can get following **Bond prices** from yield rates using the formula, $B(t_0, t_i) = \frac{100}{(1+y_i)^{\delta_i}}$

Table 3: Bond Prices

Time, T_i	$Y(T_i)$	$B(0,T_i)$
0	0.04	1
0.25	0.037	0.990835
0.5	0.04	0.980392
0.75	0.043042	0.968728
1	0.044557	0.957343
1.25	0.044943	0.94681
1.5	0.044597	0.937299
1.75	0.043916	0.928631
2	0.0433	0.920302
2.25	0.043056	0.911681

We can find the **LIBOR** starting at $t = 0$ using the formula,

$$L(t,T) = F(t,T,T+\delta) = \frac{1}{\delta}\left[\frac{B(t,T) - B(t,T+\delta)}{B(t,T+\delta)}\right]$$

Table 4: Libor Rates

Time, T_i	$L_{T_i}(0)$
0	0.037
0.25	0.0426059
0.5	0.0481637
0.75	0.0475669
1	0.0445021
1.25	0.0405855
1.5	0.0373375
1.75	0.0362025
2	0.0378246
2.25	0.0399637

We try to find the **implied volatility**, using above data of forward rates, and Black's Formula. In the following table, $V(T_{i+1})$ represents implied volatility starting at time 0 and maturing at T_{i+1}.

Table 5: Implied Volatilities

Maturity T_{i+1}	Implied Volatilities $V(T_{i+1})$
0.2500	.0625
0.5000	.0625
0.7500	.125
1.0000	.125
1.2500	.0625
1.5000	.0625
1.7500	.0625
0.0000	.03125

Assuming Stationary property, using data of implied volatilities and using following relation between implied volatilities and instantaneous volatility of LIBOR,

$$\sigma\ (T_n) = \sqrt{\frac{T_n V_n}{\delta} - \sum_{i=1}^{n-1} \sigma^2(T_i)}$$

We get following **volatility of LIBOR.**

Table 6: Volatility of LIBOR

Maturity	Start 0	0.25	0.5	0.75	1	1.25	1.5	1.75	2
0.2500	0.2500								
0.5000	0.2500	0.2500							
0.7500	0.5000	0.2500	0.2500						
1.0000	0.3536	0.5000	0.2500	0.2500					
1.2500	0.4330	0.3536	0.5000	0.2500	0.2500				
1.5000	0.2500	0.4330	0.3536	0.5000	0.2500	0.2500			
1.7500	0.2500	0.2500	0.4330	0.3536	0.5000	0.2500	0.2500		
2.0000	0.4330	0.2500	0.2500	0.4330	0.3536	0.5000	0.2500	0.2500	

.5000 highlighted above means the volatility of LIBOR starting at t=0.5 years (6 months) for time 1.25 years (15 months) to 1.5 years (18 months)

4.1. Under Spot Measure

Using above volatilities and by simultaneously generating LIBOR, we can get following drift component of LIBOR.

Drift of LIBOR under spot measure can be generated using the following formula:

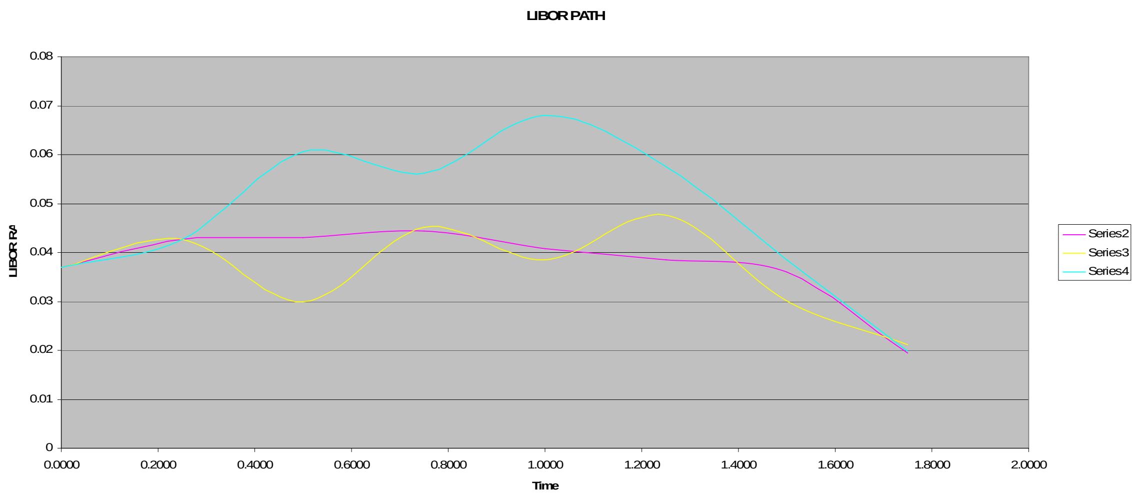
$$\hat{\mu}_n(\hat{L}(t_i), t_i) = \sum_{j=y(t)}^{n} \frac{\delta_j \hat{L}_j(t_i)\sigma_n(t_i)\sigma_j(t_i)}{1+\delta_j \hat{L}_j(t_i)}$$

Table 7: Drift of LIBOR Under Spot Measure

Maturity	Start 0	0.25	0.5	0.75	1	1.25	1.5	1.75
0.25	0.0006							
0.5	0.0012	0.00066						
0.75	0.0054	0.0014	0.00059					
1	0.0053	0.00577	0.00133	0.00072				
1.25	0.0086	0.00547	0.00546	0.0015	0.00073			
1.5	0.0056	0.00861	0.00513	0.00555	0.00136	0.00049		
1.75	0.0062	0.00554	0.00801	0.00508	0.00493	0.00111	0.00058	
2	0.0124	0.00584	0.00492	0.00711	0.00409	0.00332	0.00082	0.00018

We can generate LIBOR using the following formula and plot few paths as following:

$$\hat{L}_n(t_{i+1}) = \hat{L}_n(t_i)\exp[\{\mu_n(\hat{L}_n, t_i)\, t_i - \frac{1}{2}\sigma_n(t_i)^2\}(t_{i+1}-t_i) + \sqrt{t_{i+1}-t_i}\;\sigma_n(t_i) Z_{i+1}]$$



4.1.1. Payoff of Interest Rate Derivatives under Spot Measure

Arbitrage free price of Cap at time t=0, strike price = 4%, and face value =$ 1 is 0.046 with 95% confidence interval (0.0443, 0.0477).

Arbitrage free price of 1x1 Swaption at time t=0, strike price = 4%, and face value =$ 1 is 0.0022 with 95% confidence interval (0.0019, 0.0025).

4.2. Under Forward Measure

Similarly, we can use the above of forward rates and volatility of Libor to generate drift of LIBOR under forward measure, by simultaneously generating LIBOR, using formula,

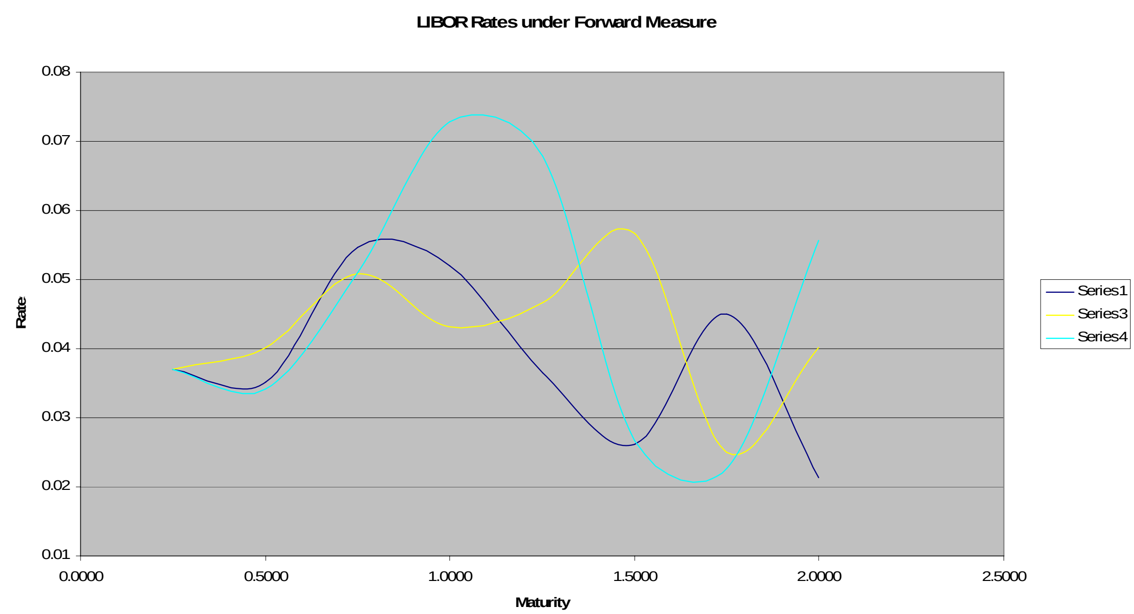
$$\mu_n(\hat{L}(t_i), t_i) = -\sum_{j=n+1}^{m} \frac{\delta_j \hat{L}_j(t_i)\sigma_n(t_i)\ \sigma_j(t_i)}{1+\delta_j \hat{L}_j(t_i)}$$

Table 8: Drift of LIBOR Under Spot Measure

Maturity	Start 0	0.25	0.5	0.75	1	1.25	1.5
0.25	-0.0066						
0.5	-0.0059	-0.0056					
0.75	-0.0088	-0.0049	-0.0046				
1	-0.0048	-0.0068	-0.004	-0.0041			
1.25	-0.0038	-0.0034	-0.0052	-0.0034	-0.0023		
1.5	-0.0015	-0.0023	-0.0024	-0.0042	-0.0017	-0.0017	
1.75	-0.001	-0.0007	-0.0013	-0.0018	-0.0014	-0.0012	-0.0005

We can generate **LIBOR** using the following formula and plot few paths:

$$\hat{L}_n(t_{i+1}) = \hat{L}_n(t_i)\exp[\{\mu_n(\hat{L}_n, t_i)t_i - \frac{1}{2}\sigma_n(t_i)^2\}(t_{i+1} - t_i) + \sqrt{t_{i+1} - t_i}\,\sigma_n(t_i)Z_{i+1}]$$



4.2.1. Payoff of Interest Rate Derivatives under Forward Measure

Arbitrage free price of Cap at time t=0, strike price = 4%, and face value =$ 1 is 0.043 with 95% confidence interval (0.037, 0.047).

Arbitrage free price of 1x1 Swaption at time t=0, strike price = 4%, and face value =$ 1 is 0.0019 with 95% confidence interval (0.00171, 0.000214).

4.3. Under Spot Measure after Martingale Discretization

Again using above data for bond prices or forward LIBOR rate starting at t = 0, we can find the following deflated bond prices under spot measure using formula:

$$\hat{D}_n(t_i) = \prod_{j=0}^{n-1} \frac{1}{1+\delta_j \hat{L}_j(T_i)}$$

Table 9: Deflated Bond Prices Under Spot Measure

Time (T_i) years	$D_{T_i}(0)$
0	1
0.25	0.990834778
0.5	0.980392157
0.75	0.968727786
1	0.957343311
1.25	0.946809551
1.5	0.937299353
1.75	0.928631169
2	0.920301859

Using above data for deflated bond price, Volatilities of LIBOR, and discretized deflated bond price equation, we can generate: volatility of deflated bond prices, deflated bond prices, and then record LIBOR Rates simultaneously.

We can generate deflated bond prices under spot measure using Euler's scheme for $LogD_{n+1}$:

$$\hat{D}_n(t_{i+1}) = \hat{D}_n(t_i)\exp[-\frac{1}{2}\hat{V}_{n+1}(t_i)^2(t_{i+1}-t_i) + \sqrt{t_{i+1}-t_i}\,\hat{V}_{n+1}(t_i)\ Z_{i+1}$$

$$\text{with } \hat{V}_{n+1}(t_i) = \sum_{j=y(t_i)}^{n} \left(\frac{\hat{D}_{j+1}(t_i)}{\hat{D}_j(t_i)} - 1\right)\sigma_j(t_i)$$

For one of the LIBOR Paths, we get following **Volatility of Deflated Bond Prices**.

Table 10: Volatility of Deflated Bond Prices

Maturity	Start 0	0.25	0.5	0.75	1	1.25	1.5	1.75
0.2500	0.0000							
0.5000	-0.0026	0.0000						
0.7500	-0.0056	-0.0038	0.0000					
1.0000	-0.0115	-0.0069	-0.0033	0.0000				
1.2500	-0.0154	-0.0125	-0.0074	-0.0028	0.0000			
1.5000	-0.0197	-0.0199	-0.0210	-0.0078	-0.0051	0.0000		
1.7500	-0.0220	-0.0216	-0.0211	-0.0163	-0.0082	-0.0037	0.0000	
2.0000	-0.0243	-0.0227	-0.0276	-0.0183	-0.0111	-0.0077	-0.0019	0.0000

For one of the LIBOR Paths, we get **following Deflated Bond Prices**

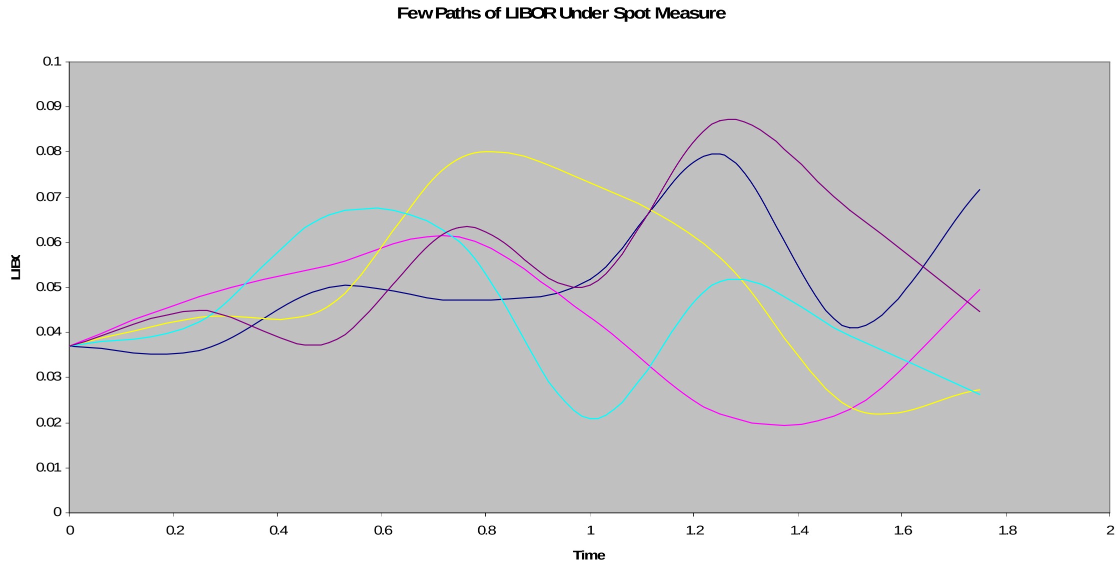
Table 11: Deflated Bond Price

Maturity	Start 0	0.25	0.5	0.75	1	1.25	1.5	1.75	2
0	1								
0.2500	0.9908	0.9908							
0.5000	0.9804	0.9819	0.9819						
0.7500	0.9687	0.9669	0.9705	0.9705					
1.0000	0.9573	0.9548	0.9576	0.9560	0.9560				
1.2500	0.9468	0.9442	0.9419	0.9454	0.9464	0.9464			
1.5000	0.9373	0.9246	0.9164	0.9265	0.9269	0.9300	0.9300		
1.7500	0.9286	0.9209	0.9161	0.9106	0.9155	0.9164	0.9150	0.9150	
2.0000	0.9203	0.9168	0.9023	0.9057	0.9103	0.9014	0.9079	0.9078	0.9078

The highlighted column were already calculated and shown in Table 9

We can generate **LIBOR using** the following formula and plot few paths as following:

$$\hat{L}_n(t_i) = \frac{1}{\delta_n} \frac{\hat{D}_n(t_i) - \hat{D}_{n+1}(t_i)}{\hat{D}_{n+1}(t_i)}$$



4.3.1. Payoff of Interest Rate Derivatives, Cap and Swaption under Spot Measure:

Arbitrage free price of Cap at time t=0, strike price = 4%, and face value =$ 1 is 0.047 with 95% confidence interval (0.047, 0.0488).

Arbitrage free price of 1x1 Swaption at time t=0, strike price = 4%, and face value =$ 1 is 0.0019 with 95% confidence interval (0.0017, 0.00198).

5. *Conclusion*

Comparison of the Results: We compare the arbitrage free prices of Swaptions and Caps under spot and forward measure before and after martingale discretizations(MD).

Table 12: Arbitrage Free Prices of Caps

Number of Simulations	Spot Measure		Forward Measure		Spot Measure After Martingale Discretization	
n	Price of Cap	95% confidence Interval	Price of Cap	95% confidence Interval	Price of Cap	95% confidence Interval
100	0.04575	(0.03979, 0.05171)	0.04333	(0.04472, 0.03501)	0.04711	(0.04177, 0.05246)
1000	0.04602	(0.04434, 0.04769)	0.04629	(0.04470, 0.04787)	0.04869	(0.04687, 0.05052)
10000	0.04601	(0.04549, 0.04655)	0.04609	(0.04579, 0.04679)	0.04825	(0.04767, 0.04882)

Table 13: Arbitrage Free Prices of Swaption

Number of Simulations	Spot Measure		Forward Measure		Spot Measure After Martingale Discretization	
n	Price of Swaption	95% confidence Interval	Price of Swaption	95% confidence Interval	Price of Swaption	95% confidence Interval
100	0.0027	(0.001627435, 0.003773)	0.002094	(0.001858, 0.003773)	0.001582	(0.000861, 0.002304)
1000	0.002989	(0.002601648, 0.003198)	0.001926	(0.001709, 0.002143)	0.001854	(0.001603, 0.002105)
10000	0.002286	(0.002105794, 0.002294)	0.001829	(0.001179, 0.002478)	0.001815	(0.001737, 0.001893)

We can observe from table that prices of the derivative are approximately same under all the measures for Cap and Swaption. Changing measure and doing martingale discretization was for convenience, accuracy, and availability of data such as bond prices.

Financial fixed income products are based on interest rate dynamics. In other words, as Interest rate derivatives are roughly divided into the two submarkets, Caps and Swaptions, having ability to price Caps and Swaptions, we can find arbitrage free prices of other financial derivatives in the market.

References

Anderson, L. and J. Andreasen, *Volatility skews and extensions of the Libor market model,* Applied Mathematical Finance, (2000).

Ang, A. and M. Sherris, *Interest Rate Risk Management: Developments in Interest Rate Term Structure Modeling for Risk Management and Valuation of Interest-Rate-Dependent Cash Flows*, North American Actuarial Journal, (1995).

Black, F., E. Derman, and W. Toy, *A One-Factor Model of Interest Rates and Its Applications to Treasury Bond Options*, Financial Analyst Journal, (1990).

Bjork, T. *Arbitrage Theory in Continuous Time* (second edition), Oxford University Press, New York, (2004).

Brigo, D., F. Mercurio, and M. Morono, *The LIBOR model dynamics: Approximations, calibration and diagnostics*, The European Journal of Operational *Research*, (2003).

Glasserman, P. and X. Zhao, *Arbitrage-free discretization of lognormal forward Libor rates an swap rate models*, Finance and Stochastics, (1999).

Glasserman, P. *Monte Carlo Methods in Financial Engineering*, Springer-Verlag, New York, (2004).

Hull, J. C. *Options, Futures and Other Derivatives* (sixth edition), Prentice Hall, New Jersey, (2005).

Jackson, M. and M. Staunton, *Advanced modeling in Finance using Excel and VBA*, John Wiley & Sons, England, (2001).

Nefti, S.N. *An Introduction to the Mathematics of Financial Derivatives* (second edition), Academic Press, California, (2000).

6. *Appendix*

Code for Implied Volatilities V_n Using Caplet Prices

```
Function IMPVOL()
'       Calculates Implied Volatilities Using Black's Formula and matching term structure
'       Number of time intervals for which data were given is i_last - i_initial
'       Calculating Caplet Price C2, using by entering dummy implied volatility V1, Forward rates, and Black's
'       formula, match C2 with Caplet price C1 from market by adjusting volatility
'       t_i and t_i+1 are entered in Column 3 and Column 4
'       delta is time interval
'       L stands for LIBOR rates at time t=0 between time interval t_i to t_i+1
'       K represents Strike Price

        i_initial = Cells(15, 2)
        i_last = Cells(22, 2)
        For i = i_initial To i_last
                delta = Cells(14 + i, 4) - Cells(14 + i, 3)
                L = Cells(29 + i, 2)
                Sum = 0
                For m = 1 To 1
                        K = Cells(29 + i, 2 + m)
                        'Calculating Payoff from Market Price
                        C1 = delta * WorksheetFunction.Max(L - K, 0)
                        Cells(53 + i, 3) = C1
'                       Calculating Payoff Using Black's Formula and Guess Volatility of LIBOR as V1. Also Taking
'                       Least Value v2 and max value v3 for volatility
                        V1 = 0.5
                        V2 = 0
                        V3 = 3
                        BDOne = (Log(L / K) + ((V1 ^ 2) * delta * 0.5)) / (V1 * Sqr(delta))
                        BDTwo = BDOne - (V1 * Sqr(delta))
                        NDOne = Application.NormSDist(Abs(BDOne))
                        NDTwo = Application.NormSDist(Abs(BDTwo))
                        C2 = delta * ((L * NDOne) - (K * NDTwo))
                        Cells(53 + i, 1) = C2
                        Do While Abs(C1 - C2) > 0.0000001
                                If C1 > C2 Then
                                        V2 = V1
                                        V1 = (V2 + V3) / 2
                                ElseIf C1 < C2 Then
                                        V3 = V1
                                        V1 = (V2 + V3) / 2
                                End If
                                BDOne = (Log(L / K) + ((V1 ^ 2) * delta * 0.5)) / (V1 * Sqr(delta))
                                BDTwo = BDOne - (V1 * Sqr(delta))
                                NDOne = Application.NormSDist(BDOne)
                                NDTwo = Application.NormSDist(BDTwo)
                                C2 = delta * ((L * NDOne) - (K * NDTwo))
                        Loop
                        Cells(53 + i, 2) = C2
                        Sum = Sum + V1
                        Cells(41 + i, 2 + m) = V1
                Next m
        Next i
End Function
```

Code for Generating Forward Libor Rates Using Libor Market Model and Pricing Caps and Swaptions under Spot measure

```
Function LIBORONE()
'           Generates Libor Rates and Calculates Prices of Caps and Swaptions
'           kk is Strike price for Cap
'           KKK is Strike price for swaption
'           del is time interval
'           m is Number of Implementations
'           Payoff of Swaption is soprice, and we set initial value for soprice as 0
'           Payoff of Cap is pay_cap2, we set initial value for Cap as 0

            kk = Cells(48, 7)
            KKK = Cells(78, 7)
            del = 0.25
            soprice = 0
            pay_cap2 = 0
            m = 1000
            For p = 1 To m
                        For j = 1 To 8
'                                   Calculating drift for LIBOR that agrees with Euler
                                    mu = 0
                                    z = 1
                                    For i = j To 8
                                                Cells(53 + i, 2 + j) = mu + (del * Cells(6 + z, 9) * Cells(6 + z, 9) * Cells(66 + i, 2 +
                                                j) / (1 + Cells(66 + i, 2 + j) * del))
                                                mu = Cells(53 + i, 2 + j) * Cells(7 + z, 9) / Cells(6 + z, 9)
                                                z = z + 1
                                    Next i
                                    z = 1
'                                   Generating LIBOR (agree with Euler)
'                                   First generate LIBOR, then Take Average
                                    For k = j + 1 To 8
                                                W = Sqr(del) * Cells(6 + z, 9) * WorksheetFunction.NormSInv(Rnd())
                                                Cells(66 + k, 3 + j) = Cells(66 + k, 2 + j) * Exp((Cells(53 + k, 2 + j) - (0.5 * Cells(6
                                                + z, 9) * Cells(6 + z, 9))) * del + W)
                                                z = z + 1
                                    Next k
                        Next j
'                       Setting initial value for price of cap
                        Price2 = 0
                        dis2 = 1
                        For pt = 1 To 8
'                                   Generating Path of Libor
                                    Sheet3.Cells(1 + p, 1 + pt) = Cells(66 + pt, 2 + pt)
'                                   Setting discount factor for first LIBOR
                                    dis2 = dis2 * (1 + Sheet3.Cells(1 + p, 1 + pt) * del)
'                                   Payoff of Caplet
                                    Sheet3.Cells(1 + p, 10 + pt) = WorksheetFunction.Max(Cells(66 + pt, 2 + pt) - kk, 0)
'                                   Discounting Caplet along the Libor Path and add them
                                    Price2 = (Sheet3.Cells(1 + p, 10 + pt) / dis2) + Price2
                        Next pt
                        'For swaption Xt4,t8(t4)- find L(t4,ti)'s then B(t4,ti)'s then find arbitrage free fixed rate
                        'then swaption price at t4 then discount to t =0
                        dis3 = 1
                        For so = 1 To 4
```

```
'                    Recording L(t4,ti) for i = 4,5,6,7
                    Sheet4.Cells(3 + p, 1 + so) = Cells(70 + so, 7)
                    dis3 = dis3 * (1 + del * Sheet4.Cells(3 + p, 1 + so))
                    Sheet4.Cells(3 + p, 6 + so) = 1 / dis3
            Next so
'           Calculating discounting factor for each path of libor
            Sheet4.Cells(3 + p, 12) = del * (Sheet4.Cells(3 + p, 7) + Sheet4.Cells(3 + p, 8) + Sheet4.Cells(3 + p,
            9) + Sheet4.Cells(3 + p, 10))
'           Calculating arbitrage free Fixed rate
            Sheet4.Cells(3 + p, 14) = (1 - Sheet4.Cells(3 + p, 10)) / Sheet4.Cells(3 + p, 12)
'           Calculating X =swaption
            Sheet4.Cells(3 + p, 16) = Sheet4.Cells(3 + p, 12) * WorksheetFunction.Max(Sheet4.Cells(3 + p, 14) -
            KKK, 0)
            soprice = soprice + Sheet4.Cells(3 + p, 16)
'           Price of Cap = Sum of Discounted Caplets
            Sheet3.Cells(1 + p, 20) = Price2
            pay_cap2 = pay_cap2 + Price2
    Next p
'   Average of Cap payoff at t0
    Cells(76, 4) = pay_cap2 / m
'   Average of Swaption at t4
    Sheet4.Cells(4, 18) = soprice / m
'   Average of swaption at t=0
    Sheet4.Cells(4, 20) = Sheet4.Cells(4, 18) * 0.92
    Cells(78, 4) = Sheet4.Cells(4, 20)
End Function
```

Code for Generating Forward Libor Rates Using Libor Market Model and Pricing Caps and Swaptions under Forward Measure

```
Function LIBORTWO()
'   Generates Libor Rates and Calculates Prices of Caps and Swaptions
'   kk is Strike price for Cap
'   KKK is Strike price for swaption
'   del is time interval
'   m is Number of Implementations
'   Payoff of Swaption is soprice, and we set initial value for soprice as 0
'   Payoff of Cap is pay_cap, we set initial value for Cap as 0

    kk = Cells(54, 7)
    KKK = Cells(56, 7)
    del = 0.25
    m = 1000
    soprice = 0
    pay_cap = 0
    For p = 1 To m
            For j = 1 To 8
'                   Calculating drift for another LIBOR that agrees with Euler
                    Cells(39, 2 + j) = 0
                    mu = 0
                    z = 1
                    For i = j To 7
                            Cells(39 - z, 2 + j) = mu - (del * Cells(26 + 1 - z, 2 + j) * Cells(26 - z, 2 + j) *
                            Cells(52 + 1 - z, 2 + j) / (1 + Cells(52 + 1 - z, 2 + j) * del))
                            mu = Cells(39 - z, 2 + j) * Cells(26 - 1 - z, 2 + j) / Cells(26 - z, 2 + j)
                            z = z + 1
```

```
                    Next i
'                   Generating LIBOR (agree with Euler) then take average
                    For k = j + 1 To 8
                        W = Sqr(del) * Cells(4 + z, 5) * Application.NormSInv(Rnd())
                        Cells(44 + k, 3 + j) = Cells(44 + k, 2 + j) * Exp((Cells(31 + k, 2 + j) - (0.5 * Cells(4
                        + z, 5) * Cells(4 + z, 5))) * del + W)
                        z = z + 1
                    Next k
                Next j
'               Setting initial value for price of cap
                Price = 0
                dis1 = 1
                For pt = 1 To 8
'                   Path of Libor
                    Sheet2.Cells(1 + p, 1 + pt) = Cells(44 + pt, 2 + pt)
                    dis1 = dis1 * (1 + Sheet2.Cells(1 + p, 1 + pt) * del)
'                   Payoff of Caplet
                    Sheet2.Cells(1 + p, 10 + pt) = WorksheetFunction.Max(Cells(44 + pt, 2 + pt) - kk, 0)
'                   Discounting Caplet along the Libor Path and add them for Calculating Cap
                    Price = (Sheet2.Cells(1 + p, 10 + pt) / dis1) + Price
                Next pt
'               For swaption Xt4,t8(t4)- find L(t4,ti)'s then B(t4,ti)'s then find arbitrage free fixed rate
'               then swaption price at t4 then discount to t =0
                dis3 = 1
                For so = 1 To 4
'                   Recording L(t4,ti) for i = 4,5,6,7
                    Sheet4.Cells(3 + p, 1 + so) = Cells(48 + so, 7)
                    dis3 = dis3 * (1 + del * Sheet4.Cells(3 + p, 1 + so))
                    Sheet4.Cells(3 + p, 6 + so) = 1 / dis3
                Next so
'               Calculating discounting factor for each path of Libor
                Sheet4.Cells(3 + p, 12) = del * (Sheet4.Cells(3 + p, 7) + Sheet4.Cells(3 + p, 8) + Sheet4.Cells(3 + p,
                9) + Sheet4.Cells(3 + p, 10))
'               Calculating arbitrage free Fixed rate
                Sheet4.Cells(3 + p, 14) = (1 - Sheet4.Cells(3 + p, 10)) / Sheet4.Cells(3 + p, 12)
'               Calculating X =swaption
                Sheet4.Cells(3 + p, 16) = Sheet4.Cells(3 + p, 12) * WorksheetFunction.Max(Sheet4.Cells(3 + p, 14) -
                KKK, 0)
                soprice = soprice + Sheet4.Cells(3 + p, 16)
'               Price of Cap = Sum of Discounted Caplets
                Sheet2.Cells(1 + p, 20) = Price
                pay_cap = pay_cap + Price
        Next p
'       Average of Swaption at t4
        Sheet4.Cells(4, 18) = soprice / m
'       Average of swaption at t=0
        Sheet4.Cells(4, 20) = Sheet4.Cells(4, 18)
        Cells(56, 4) = Sheet4.Cells(4, 20)
'       Average of Cap payoff at t0
        Cells(54, 4) = pay_cap / m
End Function
```

Code for Generating Forward Libor Rates and Pricing Caps and Swaptions Using Libor Market Model under Spot Measure After Martingale Discretization

```
Function LIBORTHREE()
'       Generates Libor Rates and Calculates Prices of Caps and Swaptions
'       kk is Strike price for Caps and Swaptions
'       delta is time interval
'       sim is Number of Implementations
'       Payoff of Swaption is soprice, and we set initial value for soprice as 0
'       Payoff of Cap is pay_cap, we set initial value for Cap as 0
        kk =
        KKK = Cells(56, 7)
        delta = 0.25
        sim = 1000
        soprice = 0
        pay_cap = 0
        For j = 1 To sim
                For m = 1 To 8
'                       Generating Vol of Deflated Bond
                        Cells(4 + m, 1 + m) = 0
                        For i = m To 7
                                term = 1 - (Cells(18 + i, 1 + m) / Cells(17 + i, 1 + m))
                                mx = WorksheetFunction.Max(0, term)
                                Cells(5 + i, 1 + m) = Cells(4 + i, 1 + m) - (Cells(4 + i, 12 + m) *
                                WorksheetFunction.Min(1, mx))
                        Next i
'                       Generating deflated bond
                        For i = m To 7
                                If i = m Then
'                                       If i = m then diagonal values
                                        W = WorksheetFunction.NormSInv(Rnd())
                                        Cells(18 + i, 2 + m) = Cells(18 + i, 1 + m) * Exp(-(Cells(5 + i, 1 + m) *
                                        Cells(5 + i, 1 + m) * delta / 2) + (Sqr(delta) * Cells(5 + i, 1 + m) * W))
                                Else
'                                       Else they r not diagonal values and we want values to decrease for
'                                       fixed start rate and wrt incrasing maturity
                                        W = WorksheetFunction.NormSInv(Rnd())
                                        D = Cells(18 + i, 1 + m) * Exp(-(Cells(5 + i, 1 + m) * Cells(5 + i, 1 + m)
                                        * delta / 2) + (Sqr(delta) * Cells(5 + i, 1 + m) * W))
                                        Do While D > Cells(17 + i, 2 + m)
                                                W = WorksheetFunction.NormSInv(Rnd())
                                                D = Cells(18 + i, 1 + m) * Exp(-(Cells(5 + i, 1 + m) * Cells(5 +
                                                i, 1 + m) * delta / 2) + (Sqr(delta) * Cells(5 + i, 1 + m) * W))
                                        Loop
                                        Cells(18 + i, 2 + m) = D
                                End If
                        Next i
'                       Generating LIBOR Rates
                        For i = m To 7
                                dn2 = Cells(18 + i, 2 + m)
                                dn1 = Cells(17 + i, 2 + m)
                                Cells(30 + i, 2 + m) = (dn1 - dn2) / (delta * dn2)
                        Next i
                Next m
                For i = 1 To 8
                        Sheet2.Cells(1 + j, 1 + i) = Cells(29 + i, 1 + i)
                Next i
```

```
                    dis3 = 1
                    For so = 1 To 4
'                        Recording L(t4,ti) for i = 4,5,6,7
                        Sheet3.Cells(1 + j, 1 + so) = Cells(33 + so, 6)
                        dis3 = dis3 * (1 + delta * Sheet3.Cells(1 + j, 1 + so))
                        Sheet3.Cells(1 + j, 10 + so) = 1 / dis3
                    Next so

‘                   Calculating Price of Cap
                    Price = 0
                    dis1 = 1
                    For pt = 1 To 8
'                        Payoff of Caplet
                        Sheet2.Cells(1 + j, 10 + pt) = WorksheetFunction.Max(Sheet2.Cells(1 + j, 1 + pt) - kk, 0)
                        dis1 = dis1 * (1 + delta * Sheet2.Cells(1 + j, 1 + pt))
'                        Discounting Caplet along the Libor Path and add them
                        Price = (Sheet2.Cells(1 + j, 10 + pt) / dis1) + Price
                    Next pt
'                   Price of Cap = Sum of Discounted Caplets
                    Sheet2.Cells(1 + j, 20) = Price
                    pay_cap = pay_cap + Price

'                   Calculating Price of Swaption
'                   For swaption Xt4,t8(t4)- find L(t4,ti)'s then B(t4,ti)'s then find arbitrage free fixed rate
'                   then swaption price at t4 then discount to t =0
'                   Calculating discounting factor for each path of libor
                    Sheet3.Cells(1 + j, 18) = delta * (Sheet3.Cells(1 + j, 11) + Sheet3.Cells(1 + j, 12) + Sheet3.Cells(1 + j,
                    13) + Sheet3.Cells(1 + j, 14))
'                   Calculating arbitrage free Fixed rate
                    Sheet3.Cells(1 + j, 20) = (1 - Sheet3.Cells(1 + j, 14)) / Sheet3.Cells(1 + j, 18)
'                   Calculating X =swaption
                    Sheet3.Cells(1 + j, 22) = Sheet3.Cells(1 + j, 18)*WorksheetFunction.Max(Sheet3.Cells(1 + j, 20)-kk, 0)
                    soprice = soprice + Sheet3.Cells(1 + j, 22)
          Next j
'         Average of Swaption at t4
          Sheet3.Cells(2, 24) = soprice / sim
          bond = Cells(21, 2)
'         Average of swaption at t=0
          Sheet3.Cells(4, 24) = Sheet3.Cells(2, 24) * bond
'         Average of Cap payoff at t0
          Sheet2.Cells(2, 22) = pay_cap / sim
End Function
```