FIN 516: Project

1. Data Processing

1.1 Data Collection

Our first step was to obtain market prices from Bloomberg for LIBOR deposit rates and Eurodollar Futures as well as cap and swaption volatilities on our product's pricing date – October 25, 2019. The market data was obtained from the products' pricing date (October 25, 2019) until its maturity date (April 29, 2023).

Forward rates were linearly interpolated such that we had estimates for the rate every 6 months as interest payments were made semi-annually. From the forward rates, zero-coupon bond prices were calculated using:

$$P(0, T+\tau) = \frac{1}{1+\tau \times F(0;T,T+\tau)}$$

The table below summarizes the interpolated forward rates along with their corresponding zero-coupon bond prices.

Time	Date	Forward rate	P(0,T)
T0	10/29/2019	1.9281%	1
3M	01/29/2020	1.8227%	0.995096717
6M	04/29/2020	1.6450%	0.990532881
9M	07/29/2020	1.5785%	0.986431072
1Y	10/29/2020	1.5518%	0.982467879
1Y 3M	01/29/2021	1.5313%	0.978587113
1Y 6M	04/29/2021	1.4778%	0.97485501
1Y 9M	07/29/2021	1.4299%	0.971226926
2Y	10/29/2021	1.5290%	0.967690877
2Y 3M	01/31/2022	1.5092%	0.963842966
2Y 6M	04/29/2022	1.4903%	0.960300259
2Y 9M	07/29/2022	1.4709%	0.956696327
3Y	10/31/2022	1.5273%	0.953036058
3Y 3M	01/30/2023	1.5196%	0.949370743
3Y 6M	04/28/2023	1.5121%	0.945857254

Next, at-the-money cap volatilities were obtained. Given that our callable note has a term of approximately 3.5 years, we got ATM cap volatilities with tenors from 1 to 4 years. The table below shows ATM cap volatilities over different expiries.

Tenor	Market Cap Volatility
1	0.3338
2	0.4125
3	0.4431

4	0.4592

Finally, swaption volatilities were obtained from Bloomberg. This data will be used to value swaptions and calibrate our LMM model. The table below shows the volatilities of swaptions. The rows represent the maturity of each swaption whereas the columns are the tenors of the underlying swap.

Expiry	1Yr	2Yr	3Yr	4Yr	5Yr	6Yr
1Mo	39.62	41.63	48.39	45.34	49.73	46.43
3Mo	43.15	41.77	52.7	50.48	49.56	45.32
6Mo	38.7	47.4	48.64	47.67	44.59	45.07
9Mo	45.55	46.24	46.8	46.64	45.66	44.63
1Yr	44.32	44.96	46.83	45.89	45.63	44.12
2Yr	45.59	45.62	45.26	44.48	43.27	42.46
3Yr	45.1	44.33	42.3	42.39	41.34	40.1
4Yr	43.92	42.64	40.95	39.75	38.55	38.29
5Yr	41.96	40.63	39.91	37.78	38.1	36.58

1.2 Stripping Caplet Volatilities

Our next step was to run a stripping algorithm to obtain caplet volatilities from cap volatilities. We assumed that the caps consisted of caplets which reset every 3 months with the first payment after 6 months. The value of a cap as the sum of individual caplets can be written as:

$$Cap(0, T_0, \ldots, T_n, K) = \sum_{i=0}^{n-1} \tau_i P(0, T_{i+1}) Black\left(K, F(0; T_i, T_{i+1}), \sqrt{T_i}\sigma_{T_{i+1}}^{caplet}\right)$$

where $\sigma_{T_{i+1}}^{caplet}$ is the caplet volatility for the period T_{i+1} and $Black\left(K, F(0; T_i, T_{i+1}), \sqrt{T_i}\sigma_{T_{i+1}}^{caplet}\right)$ is Black's caplet formula with strike K, forward rate $F(0; T_i, T_{i+1})$, and volatility $\sqrt{T_i}\sigma_{T_{i+1}}^{caplet}$.

The Black formula of caplet is given by the following equation:

$$Caplet(0, T, T+\tau, K) = N \times \tau \times P(0, T+\tau) \times [F(0; T, T+\tau) \times N(d_1) - K \times N(d_2)]$$

$$d_1 = \frac{\ln\left(\frac{F(0; T, T+\tau)}{K}\right) + \frac{1}{2}\sigma_{Black}^2 T}{\sigma_{Black}\sqrt{T}}$$

$$d_2 = \frac{\ln\left(\frac{F(0; T, T+\tau)}{K}\right) - \frac{1}{2}\sigma_{Black}^2 T}{\sigma_{Black}\sqrt{T}}$$

Caps are said to be at-the-money if the exercise price K is equal to the forward swap rate, $S_{0.25,T_n}(0)$. Forward swap rates were calculated using the equation:

$S_{T_0,T_n}(t) = \frac{P(t,T_0)-P(t,T_n)}{\sum_{i=0}^{n-1}\tau_i(t,T_{i+1})}$

After computing forward swap rates, we began with the shortest dated 1-year cap. Given that we had quarterly reset dates, the effective start date was $t = 0.25$. The first volatility estimate we had was ${\sigma_1}^{cap}$. However, we had 2 additional caps with maturities T = 0.5 and T = 0.75 for which we were missing volatility estimates. Therefore, we assumed that the volatility was constant over the first year. That is,

${\sigma_{0.5}}^{cap} = {\sigma_{0.75}}^{cap} = {\sigma_1}^{cap}$

For intermediate stages after the first year during which cap volatilities were missing, approximations where computed by linearly interpolating available market implied volatilities.

After computing cap prices as the sum of individual caplets, implied volatilities were evaluated for the last caplet within each cap using the Black model. With these new implied volatilities, we were able to better estimate the prices of the remaining caplets within each cap. This algorithm enabled us to extract quarterly caplet volatilities. The table below shows our findings.

Date	Caplet price, maturity T+t	IV	Black	SE
04/29/2020	0.00030687	0.3338	0.00030687	1.30104E-18
07/29/2020	0.000297731	0.3338	0.000297731	0
10/29/2020	0.000355021	0.3338	0.000355021	1.0842E-19
01/29/2021	0.000517475	0.39531149	0.000517475	5.04182E-11
04/29/2021	0.000624044	0.4281805	0.000624044	7.03154E-11
07/29/2021	0.000704806	0.46240077	0.000704805	1.09803E-09
10/29/2021	0.000792574	0.49709878	0.000792574	2.67262E-10
01/31/2022	0.000942901	0.45672568	0.000942899	1.47871E-09
04/29/2022	0.000938939	0.46998663	0.000938938	1.06449E-09
07/29/2022	0.001024652	0.48326345	0.001024652	1.85632E-10
10/31/2022	0.00111421	0.49709287	0.001114209	5.0891E-10
01/30/2023	0.001170816	0.47844465	0.001170816	1.0305E-12
04/28/2023	0.001178089	0.48547851	0.001178089	1.98891E-10

2. BDT Model

2.1 BDT Calibration

Once we extracted caplet volatilities, we had to calibrate our short rate model to cap market prices. We chose the BDT as our short rate model because …

First, we constructed a tree with 42 monthly time steps. That is, $\Delta t = \frac{1}{12}$. Although we only had volatilities $\sigma_{Black}(T_k)$ associated with each of the reset dates T_k, we wished to specify the volatilities at each time step. We let steps i_k correspond to the reset dates of the caplets such that $i_k \times \Delta t = T_k$. Then, time steps up to $(i_2 - 1) \times \Delta t$, we set the volatilities to $\sigma_{Black}(T_1)$. That is,

$\sigma(0), \dots, \sigma(i_2 - 1) = \sigma_{Black}(T_1)$

For every time step thereafter,

$\sigma(i_k), \dots, \sigma(i_{k+1} - 1) = \sigma_{Black}(T_k)$

Next, based on these volatilities, we chose $U(i)$ values which matched the zero-coupon bond prices. Logarithmic interpolations were performed to estimate missing zero-coupon bond prices. To find $U(i)$ values we needed to build $\hat{\pi}$ and r trees. We began the trees by setting $\hat{\pi}(0,0) = 1$ and found $r(0,0)$ by solving $\hat{P}(1) = \hat{\pi}(0,0)e^{-r(0,0)\Delta t}$ where $\hat{P}(1)$ is the market zero-coupon bond price at the first time step. Then, we let $\hat{\pi}(1,0) = \hat{\pi}(1,1) = \frac{1}{2}\hat{\pi}(0,0)e^{-r(0,0)\Delta t}$. Finally, $U(1)$ was adjusted to the market price $\hat{P}(2)$ such that $\hat{P}(2) = \sum_{j=0}^{1} \hat{\pi}(1,j)\exp\left(-U(1)e^{(2j-1)\sigma(1)\sqrt{\Delta t}}\Delta t\right)$. $r(1,j)$ values were determined using $r(i,j) = U(i)e^{(2j-i)\sigma(i)\sqrt{\Delta t}}$ where $0 \le j \le i$. New $\hat{\pi}(2,j)$ values were calculated using $\hat{\pi}(i,0) = \frac{1}{2}\hat{\pi}(i-1,0)e^{-r(i-1,0)\Delta t}$, $\hat{\pi}(i,j) = \frac{1}{2}\hat{\pi}(i-1,j)e^{-r(i-1,j)\Delta t} + \frac{1}{2}\hat{\pi}(i-1,j-1)e^{-r(i-1,j-1)\Delta t}$, and $\hat{\pi}(i,i) = \frac{1}{2}\hat{\pi}(i-1,i-1)e^{-r(i-1,i-1)\Delta t}$. $U(2)$ was approximated such that $\hat{P}(i+1) = \sum_{j=0}^{1} \hat{\pi}(i,j)\exp\left(-U(i)e^{(2j-i)\sigma(i)\sqrt{\Delta t}}\Delta t\right)$. The process was then repeated until i was equal to the number of time steps in the tree.

From the short rate tree, we were able to price caplets and see how closely they matched Black prices. Piecewise constant volatilities were then modified to better match those prices. New volatilities meant that we had to readjust $U(i)$ values.

Our algorithm allowed us to calibrate to market zero-coupon bond and caplet prices. Our results are summarized in the table below.

n	**Time**	**P(0, T)**	**New P(0, T)**	**% Error**
0	0	1	1	0
3	0.25	0.995096717	0.995097653	-0.0001%
6	0.5	0.990532881	0.990537281	-0.0004%
9	0.75	0.986431072	0.986438257	-0.0007%
12	1	0.982467879	0.982478871	-0.0011%
15	1.25	0.978587113	0.978606972	-0.0020%
18	1.5	0.97485501	0.974873794	-0.0019%
21	1.75	0.971226926	0.971248195	-0.0022%
24	2	0.967690877	0.967719765	-0.0030%
27	2.25	0.963842966	0.963896786	-0.0056%
30	2.5	0.960300259	0.960338911	-0.0040%
33	2.75	0.956696327	0.956734768	-0.0040%
36	3	0.953036058	0.953091157	-0.0058%
39	3.25	0.949370743	0.949574569	-0.0215%
42	3.5	0.945857254	0.946028192	-0.0181%

As we can see from the percentage error term, the errors between both the zero coupon bond price and the caplet price are small, which indicates a good calibration result of the BDT model.

Time	Market caplet value	Tree value	%Error
0.00	N/A	N/A	
6.00	0.000306870	0.00030687	0.00000%
9.00	0.000297731	0.00029773	0.00000%
12.00	0.000355021	0.00035502	0.00005%
15.00	0.000517475	0.00051747	0.00001%
18.00	0.000624044	0.00062404	0.00011%
21.00	0.000704806	0.00070481	0.00000%
24.00	0.000792574	0.00079257	0.00001%
27.00	0.000942901	0.0009429	0.00001%
30.00	0.000938939	0.00093894	0.00001%
33.00	0.001024652	0.00102465	0.00007%
36.00	0.001114210	0.00111421	0.00015%
39.00	0.001170816	0.00117082	0.00005%
42.00	0.001178089	0.00117809	0.00000%

##MAKE COMMENTS ON RESULTS, LIMITATIONS OF THE BDT MODEL, HOW IT COULD BE DIFFERENT USING HW MODEL##

2.2 European Swaption Valuation

Using the swaption volatilities from Bloomberg, we chose to value swaptions for which the sum of the maturity and tenor was less than or equal to the maturity of our selected structured product. The table below shows the volatilities of the ten swaptions we chose to value based on their maturities and tenors.

Maturity/Tenor	1Yr	2Yr	3Yr
3Mo	43.15	41.77	52.7
6Mo	38.7	47.4	48.64
9Mo	45.55	46.24	
1Yr	44.32	44.96	

Our first step in valuing the swaptions was to calculate the swap rate. The swap rates were calculated with:

$$S_{T_0,T_n}(t) = \frac{P(t,T_0) - P(t,T_n)}{\sum_{i=0}^{n-1} \tau_i P(t,T_{i+1})}$$

where T_0 is the maturity of the swaption, T_n is equal to the sum of the maturity and the tenor, and $P(t,T_0)$ is the value of a zero-coupon bond from t until T_0.

We built the coupon bond tree for each swaption using the BDT rate tree obtained from the calibration. Coupons were paid quarterly, the annual coupon rate was equal to the swap rate, and the maturity of the bond was equal to the maturity of the underlying swap of the swaption (sum of the maturity of the swaption and the tenor). A different coupon bond had to be valued for each of the swaptions. Coupon bond trees were built by discounting the notional back to the maturity date of the swaption (T_0) while adding quarterly coupons. At each of the bonds' maturity, payoffs were calculated using: max $(1 - CB_{i,j}, 0)$. The discounted payoffs were calculated by multiplying the payoff by the Arrow-Debreu price at that maturity state.

For comparison, Black prices for each of the European swaptions were calculated using:

$$Swaption(T_0, T_n) = N \times \sum_{i=0}^{n-1} \tau_i P(t, T_{i+1}) \times [S_{T_0,T_n}(0)N(d_1) - KN(d_2)]$$

$$d_1 = \frac{\ln\left(\frac{S_{T_0,T_n}(0)}{K}\right) + \frac{1}{2}\sigma_{S,black}^2 T}{\sigma_{S,black}\sqrt{T}}$$

$$d_2 = \frac{\ln\left(\frac{S_{T_0,T_n}(0)}{K}\right) - \frac{1}{2}\sigma_{S,black}^2 T}{\sigma_{S,black}\sqrt{T}}$$

Swaption values computed using price trees and Black's formula are summarized in the table below.

Maturity	**Tenor**	**Volatility**	**Black price**	**Tree price**	**Raw Error**	**Percent Error**
0.25	1	0.4315	0.0014183	0.0011287	0.00028956	20.42%
0.25	2	0.4177	0.0025993	0.0025289	7.04E-05	2.71%
0.25	3	0.527	0.0047929	0.003939	0.00085394	17.82%
0.5	1	0.387	0.0017062	0.0015672	0.00013909	8.15%
0.5	2	0.474	0.0040237	0.0034633	0.00056031	13.93%
0.5	3	0.4864	0.0060999	0.0053954	0.00070452	11.55%
0.75	1	0.4555	0.0023773	0.002069	0.00030831	12.97%
0.75	2	0.4624	0.0047188	0.0044049	0.00031391	6.65%
1	1	0.4432	0.0025915	0.0025738	1.78E-05	0.69%
1	2	0.4496	0.0052349	0.0053077	7.28E-05	1.39%

In the above table, the raw error is the absolute difference between the black price and the tree price while the percent error is the raw error divided by the black price (percentage deviation from the black price). We observed that in some cases the percent error is quite large. We deduced that our large errors could be due to:

- The assumptions that we had made during the calibration process. The BDT model assumes that forward rates are perfectly correlated. This assumption does not hold in the real world. Although swaption prices are correlated with forward rates of similar maturity, the level of correlation might fluctuate from year to year. The changing correlations might result to difference between black and tree valuations.

- The proximity of the maturity to the today's date. We noticed that in the case of the first swaption the prices were the lowest while the percent error was the largest (neared 20%). In the case of small prices, even small deviations can result in large percent errors.
- The length of the tenor. We noticed that in the case of swaptions with longer tenors, the error tended to be larger. This could be because cap information was not as reliable over longer time periods than over shorter time periods. This effect could have impacted our calibration results and thus lead to larger valuation errors.
- Large volatility jumps.

2.3 Product Valuation Using BDT

a. Product Overview

https://www.sec.gov/Archives/edgar/data/1114446/000091412119002870/ub54248280-424b2.htm

We chose to value product 3, the callable note issued by UBS AG London Branch. The properties of the structured product are summarized in the table below.

Pricing Date	10/25/2019
Maturity Date	04/29/2023
Interest Payment Dates	Semi-annually, from 04/29/2020 to 04/29/2023
Redemption Dates	Quarterly, from 01/29/2020 to 01/29/2023
Interest rate	2.2% until 04/29/2022; 2.5% until 04/29/2023

The timeline below highlights important dates within the duration of the note. The dates in green cells are interest payment dates while the dates in orange cells are not, but the product on both kinds of dates can be redeemed.

0	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13
10/25/19	1/29/19	4/29/19	7/29/19	10/29/19	1/29/20	4/29/20	7/29/20	10/29/20	1/29/21	4/29/21	7/29/21	10/29/21	1/29/22	4/29/22

Interest Rate = 2.2%

Interest Rate = 2.5%

b. Product Valuation Procedure

Our product was priced using a tree. We built a coupon bond tree with a call feature at every redemption date. We filled the tree backwards and used the calibrated short-rate tree from the BDT procedure for discounting.

We set the values at T_{13} (04/29/2023) to:

$$V_{42,j} = 1000 \times \left(1 + 2.5\% \times \frac{180}{360}\right) = 1012.5$$

We set values for dates on which the note can be redeemed but no interest is paid ($T_2, T_4, T_6, T_8, T_{10}, T_{12}$) to:

$$V_{i,j} = \min\left[\frac{1}{1+r_{i,j}\Delta t} \times \left(0.5V_{i+1,j+1} + 0.5V_{i,j+1}\right), 1000 \times \left(1 + CPN_{T_i} \times \frac{90}{360}\right)\right]$$

$$CPN_{T_i} = \begin{cases} 2.2\%, & when\ i = 2, 4, 6, 8, 10 \\ 2.5\%, & when\ i = 12 \end{cases}$$

We set values on interest payment dates $(T_1, T_3, T_5, T_7, T_9, T_{11})$ to:

$$V_{i,j} = \min\left[\frac{1}{1+r_{i,j}\Delta t} \times \left(0.5V_{i+1,j+1} + 0.5V_{i,j+1}\right), 1000\right] + 1000 \times CPN_{T_i} \times \frac{180}{360}$$

$$CPN_{T_i} = \begin{cases} 2.2\%, & when\ i = 1, 3, 5, 7, 9 \\ 2.5\%, & when\ i = 11 \end{cases}$$

Values on any other date were set toe:

$$V_{i,j} = \frac{1}{1+r_{i,j}\Delta t} \times \left(0.5V_{i+1,j+1} + 0.5V_{i,j+1}\right)$$

Using this algorithm, we valued the structured product at $998.11 by performing the above algorithm in the spreadsheet. Given that UBS AG priced its product at $1000, our estimation is 0.1890% away from the true value. The BDT model does a good job in valuing the product. Overall, we think that the percent error is small enough to think that the BDT model did a good job of estimating the value of the product. We think that our valuation could be improved by choosing a smaller time step. We choose our time step to be monthly, the result might become more accurate if the time step is smaller.

2.4 Bermudan Swaption Valuation

A Bermudan swaption is an option that offers the buyer the tight to take part in an interest rate swap on specified dates during the life of the option. Using our calibrated BDT short rate tree, we decided to value a 2-year Bermudan swaption on a 3-year swap issued at time 0 and paying LIBOR while receiving 1.34%. The swaption could be exercised quarterly at times 0.25, 0.50, 0.75, 1.00, 1.25, 1.50, and 1.75. The principal of the swaption is $1000.

We valued our Bermudan swaption using trees with monthly time steps. We first filled backwards a 3-year coupon bond tree.

We set the values at T_{36} to:

$$CB_{36,j} = 1000 \times (1 + 2.5\% \times 0.25) = 1006.25$$

On coupon payment dates $(T_3, T_6, T_9, T_{12}, T_{15}, T_{18}, T_{21}, T_{24}, T_{27}, T_{30}, T_{33}, T_{36})$, values in the tree were set to:

$$CB_{i,j} = \frac{1}{1+r_{i,j}\Delta t} \times \left(0.5CB_{i+1,j+1} + 0.5CB_{i,j+1}\right) + 1000 \times 2.5\% \times 0.25$$

On any other date in the tree, values were set to:

$$CB_{i,j} = \frac{1}{1 + r_{i,j}\Delta t} \times \left(0.5CB_{i+1,j+1} + 0.5CB_{i,j+1}\right)$$

Using our coupon bond tree, we were able to build a tree for our Bermudan swaption. We built the tree backwards starting at the 2-year mark.

We set values at T_{24} to:

$$V_{24,j} = max\left(CB_{24,j} - 1000 \times (1 + 2.5\% \times 0.25), 0\right)$$

On early exercise dates $(T_3, T_6, T_9, T_{12}, T_{15}, T_{18}, T_{21})$, values in the tree were set to:

$$V_{i,j} = max\left(CB_{24,j} - 1000 \times (1 + 2.5\% \times 0.25), \frac{1}{1 + r_{i,j}\Delta t} \times \left(0.5V_{i+1,j+1} + 0.5V_{i,j+1}\right)\right)$$

On any other date in the tree, values were set to:

$$V_{i,j} = \frac{1}{1 + r_{i,j}\Delta t} \times \left(0.5V_{i+1,j+1} + 0.5V_{i,j+1}\right)$$

We obtained the value of the swaption to be $4.63.

3. LIBOR Market Model

3.1 Calibration

Traditional term structure models specify a process for the instantaneous short rate or the instantaneous forward rate. On the other hand, the idea behind the LIBOR Market Model (LMM) is to model forward LIBOR rates. LMM is popular among practitioners because it models market-observable quantities and is consistent with the standard approach of pricing caps using Black's formula. Therefore, the LMM can be used to price any instrument whose payoff can be decomposed into a set of forward rates.

Given a set of dates $0 \leq T_0 < T_1 < \ldots < T_n$, the objective is to model the LIBOR rate $L(T_{i-1}, T_i)$ for each period $[T_{i-1}, T_i]$ when $i = 1, \ldots, n$. Let $L(T_{i-1}, T_i) = F(T_{i-1}, T_{i-1}, T_i) = F_i(T_{i-1})$. Like the Black model, the LMM assumes a log-normal process for each $F_i(t)$ under the T_i-forward measure $\mathbb{Q}_{T_i}$:

$$dF_i(t) = \sigma_i(t)F_i(t)dW_i^{\mathbb{Q}_{T_i}}, i = 1, \ldots, n$$

where $\sigma_i(t)$ is the time-dependent instantaneous volatility and $W_i^{\mathbb{Q}_{T_i}}$ is a Brownian motion under $\mathbb{Q}_{T_i}$. We also assume that the Brownian motions are correlated such that:

$$E\left[dW_i dW_j\right] = \rho_{ij}dt$$

where ρ_{ij} instantaneous correlation between $F_i(t)$ and $F_j(t)$. The instantaneous correlation describes the interrelations in changes in forward rates.

From the equations above, we note that changes in the LMM forward rate depend on instantaneous volatilities and correlations. Deterministic functions for the volatilities and the correlations can be calibrated using market data.

(1) Calibration of volatility function

For estimating instantaneous volatilities, we chose a function which would allow us to use five parameters to match caplet prices. Our parametric equation can be written as:

$$\sigma^2{}_{Black,i} = \frac{\Phi^2{}_i}{T_{i-1}} \int_0^{T_{i-1}} \left([b(T_{i-1} - t) + a] e^{-c(T_{i-1}-t)} + d \right)^2 dt$$

We chose to use this function as we could select Φ_i values to match the caplet volatilities precisely. The parameters used to calibrate the function to the caplet volatilities are shown in the table below.

Parameter	**Calibrated Value**
a	0. 3011
b	2.5463e-07
c	1.6758
d	-0.5465

Parameter	**Calibrated Value**
Φ_1	1.1055
Φ_2	0.9645
Φ_3	0.8783
Φ_4	0.9728
Φ_5	1.0032
Φ_6	1.0438
Φ_7	1.0905
Φ_8	0.9796
Φ_9	0.9899
Φ_{10}	1.0031
Φ_{11}	1.0194
Φ_{12}	0.9713
Φ_{13}	0.9772

The reason why we chose function 7 to calibrate volatility is that we want to generate the hump shape. Here Φ_i are all close to 1, which means that the volatilities of each forward rate are time homogeneous and just depend upon time remaining until expiry.

(2) Calibration of correlation function

With regards to the instantaneous correlations, we wanted them to always be greater than or equal to 0, indicate that the joint movement of far way rates be less correlated than those of close rates, and increase with tenors. We generated an $M \times M$ full rank correlation matrix where M was the was the number of forward rates required. Instantaneous correlations were approximated using:

$$\rho_{i,j} = exp\left(-\beta \left| T_i - T_j \right|\right)$$

where $\beta \geq 0$.

We estimated $\beta = 0.3104$. In the table below is shown the calculated correlation matrix.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	1.000000	0.925323	0.856223	0.792283	0.733118	0.678371	0.627713	0.580837	0.537462	0.497326	0.460187	0.425822	0.394023	0.364599
1	0.925323	1.000000	0.925323	0.856223	0.792283	0.733118	0.678371	0.627713	0.580837	0.537462	0.497326	0.460187	0.425822	0.394023
2	0.856223	0.925323	1.000000	0.925323	0.856223	0.792283	0.733118	0.678371	0.627713	0.580837	0.537462	0.497326	0.460187	0.425822
3	0.792283	0.856223	0.925323	1.000000	0.925323	0.856223	0.792283	0.733118	0.678371	0.627713	0.580837	0.537462	0.497326	0.460187
4	0.733118	0.792283	0.856223	0.925323	1.000000	0.925323	0.856223	0.792283	0.733118	0.678371	0.627713	0.580837	0.537462	0.497326
5	0.678371	0.733118	0.792283	0.856223	0.925323	1.000000	0.925323	0.856223	0.792283	0.733118	0.678371	0.627713	0.580837	0.537462
6	0.627713	0.678371	0.733118	0.792283	0.856223	0.925323	1.000000	0.925323	0.856223	0.792283	0.733118	0.678371	0.627713	0.580837
7	0.580837	0.627713	0.678371	0.733118	0.792283	0.856223	0.925323	1.000000	0.925323	0.856223	0.792283	0.733118	0.678371	0.627713
8	0.537462	0.580837	0.627713	0.678371	0.733118	0.792283	0.856223	0.925323	1.000000	0.925323	0.856223	0.792283	0.733118	0.678371
9	0.497326	0.537462	0.580837	0.627713	0.678371	0.733118	0.792283	0.856223	0.925323	1.000000	0.925323	0.856223	0.792283	0.733118
10	0.460187	0.497326	0.537462	0.580837	0.627713	0.678371	0.733118	0.792283	0.856223	0.925323	1.000000	0.925323	0.856223	0.792283
11	0.425822	0.460187	0.497326	0.537462	0.580837	0.627713	0.678371	0.733118	0.792283	0.856223	0.925323	1.000000	0.925323	0.856223
12	0.394023	0.425822	0.460187	0.497326	0.537462	0.580837	0.627713	0.678371	0.733118	0.792283	0.856223	0.925323	1.000000	0.925323
13	0.364599	0.394023	0.425822	0.460187	0.497326	0.537462	0.580837	0.627713	0.678371	0.733118	0.792283	0.856223	0.925323	1.000000

Using the correlation matrix ρ, we computed its square root b such that $bb^T = \rho$.

However, at the beginning, we tried to use the three-parameter model but failed since the result didn't converge even though we set the parameter range. We thought that sometimes the more complicated function might not give us better results than the simpler function. Therefore, we came back to the one-parameter model and successfully did it.

(3) Another attempt

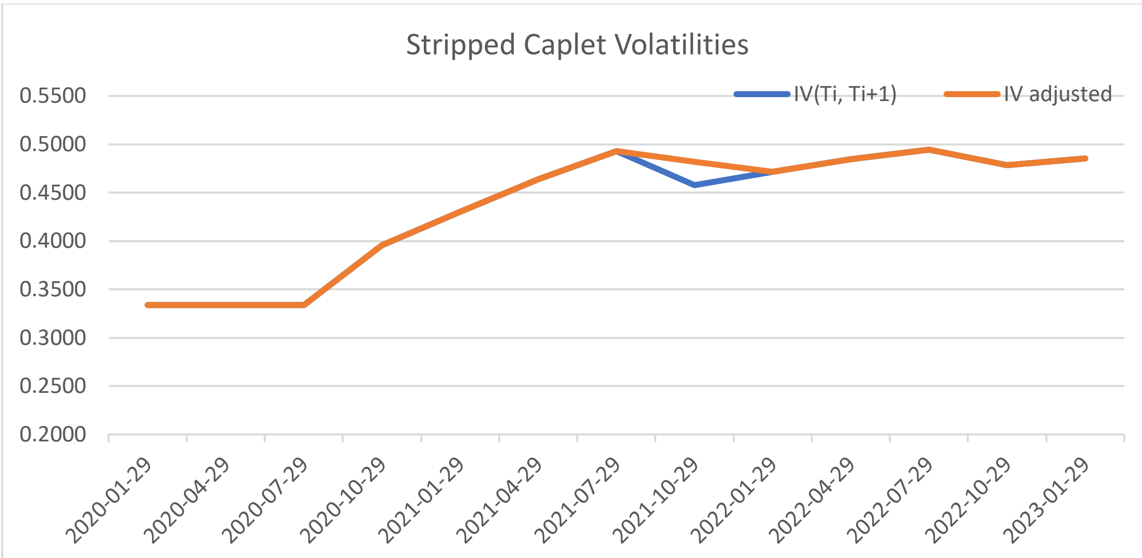
We also tried using Function 2 to calibrate volatilities. Function 2 is defined as:

$$\sigma^2{}_{Black,k} = \frac{1}{T_{k-1}} \sum_{j=1}^{k} \tau_{j-1} \eta_{k-j+1}^2$$

Since η_{k-j+1}^2 cannot be negative, $\sigma^2{}_{Black,k} T_{k-1}$ should increase as k increases. But the stripped caplet volatilities did not strictly comply with the rule above. So we manually interpolated the cells highlighted. The $\eta_i^2 s$ are also listed in the following chart. Stripped caplet volatilities and stripped caplet volatilities after adjustment are showed in the graph below.

Index	**Date**	**IV(Ti, Ti+1)**	**IV^2*Ti**	**IV adjusted**	**IV_adj^2*Ti**	**ita**
0	2019-10-29					
T0	2020-01-29	0.3338	0.0279	0.3338	0.0279	**0.3338**
T1	2020-04-29	0.3338	0.0557	0.3338	0.0557	**0.3338**
T2	2020-07-29	0.3338	0.0836	0.3338	0.0836	**0.3338**
T3	2020-10-29	0.3956	0.1560	0.3956	0.1560	**0.5384**
T4	2021-01-29	0.4304	0.2290	0.4304	0.2290	**0.5403**
T5	2021-04-29	0.4638	0.3208	0.4638	0.3208	**0.6059**
T6	2021-07-29	0.4931	0.4323	0.4931	0.4323	**0.6677**
T7	2021-10-29	0.4580	0.4171	0.4822	0.4676	**0.3763**
T8	2022-01-29	0.4712	0.4973	0.4712	0.4973	**0.3442**

T9	2022-04-29	0.4843	0.5848	0.4843	0.5848	**0.5918**
T10	2022-07-29	0.4944	0.6802	0.4944	0.6802	**0.6176**
T11	2022-10-29	0.4787	0.6859	0.4787	0.6859	**0.1510**
T12	2023-01-29	0.4854	0.7654	0.4854	0.7654	**0.5641**
T13	2023-04-29					



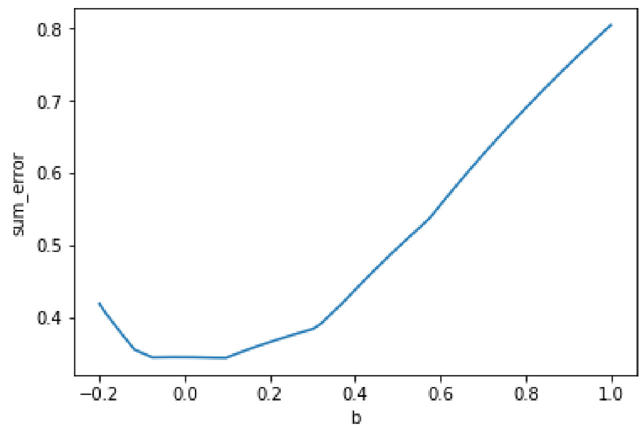
Then we calibrated the correlations related to swaptions. According to Robenato's formula, the Black vol of a swaption with a maturity of α and a tenor of (β-α) is

$$\sigma_{\alpha,\beta}^2 \approx \sum_{i,j=\alpha+1}^{\beta} \frac{\omega_i(0)\omega_j(0)F_i(0)F_j(0)\rho_{i,j}}{T_\alpha S_{\alpha,\beta}^2(0)} \int_0^\alpha \sigma_i(t)\sigma_j(t)dt, \text{ where}$$

$$\omega_k(0) = \frac{\tau_k * P(0,T_k)}{\sum_{j=\alpha+1}^{\beta} \tau_j * P(0,T_j)}$$

Replace $\int_0^\alpha \sigma_i(t)\sigma_j(t)dt$ with $\sum_{l=1}^{\alpha} \eta_{i-l+1}\eta_{j-l+1}\tau_{l-1}$, and $\rho_{i,j}$ with $exp(-b|T_i - T_j|)$.

With ηs solved in when calibrating to caps, and P(0, t)s, $F_i(0)$s and $\sigma_{\alpha,\beta}^2$ available in Bloomberg, we can solve for b. The graph below shows the relation between b and the sum of the absolute value of errors. The optimal b is 0.0952.



With the optimal b and ηs, we generated the correlation matrix and covariance matrix.

Correlation Matrix (b = 0.0952)

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	1.0000	0.9765	0.9535	0.9311	0.9092	0.8878	0.8669	0.8465	0.8266	0.8072	0.7882	0.7697	0.7516
1	0.9765	1.0000	0.9765	0.9535	0.9311	0.9092	0.8878	0.8669	0.8465	0.8266	0.8072	0.7882	0.7697
2	0.9535	0.9765	1.0000	0.9765	0.9535	0.9311	0.9092	0.8878	0.8669	0.8465	0.8266	0.8072	0.7882
3	0.9311	0.9535	0.9765	1.0000	0.9765	0.9535	0.9311	0.9092	0.8878	0.8669	0.8465	0.8266	0.8072
4	0.9092	0.9311	0.9535	0.9765	1.0000	0.9765	0.9535	0.9311	0.9092	0.8878	0.8669	0.8465	0.8266
5	0.8878	0.9092	0.9311	0.9535	0.9765	1.0000	0.9765	0.9535	0.9311	0.9092	0.8878	0.8669	0.8465
6	0.8669	0.8878	0.9092	0.9311	0.9535	0.9765	1.0000	0.9765	0.9535	0.9311	0.9092	0.8878	0.8669
7	0.8465	0.8669	0.8878	0.9092	0.9311	0.9535	0.9765	1.0000	0.9765	0.9535	0.9311	0.9092	0.8878
8	0.8266	0.8465	0.8669	0.8878	0.9092	0.9311	0.9535	0.9765	1.0000	0.9765	0.9535	0.9311	0.9092
9	0.8072	0.8266	0.8465	0.8669	0.8878	0.9092	0.9311	0.9535	0.9765	1.0000	0.9765	0.9535	0.9311
10	0.7882	0.8072	0.8266	0.8465	0.8669	0.8878	0.9092	0.9311	0.9535	0.9765	1.0000	0.9765	0.9535
11	0.7697	0.7882	0.8072	0.8266	0.8465	0.8669	0.8878	0.9092	0.9311	0.9535	0.9765	1.0000	0.9765
12	0.7516	0.7697	0.7882	0.8072	0.8266	0.8465	0.8669	0.8878	0.9092	0.9311	0.9535	0.9765	1.0000

Covariance Matrix

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0.0279	0.0272	0.0266	0.0418	0.0410	0.0449	0.0483	0.0266	0.0237	0.0399	0.0406	0.0097	0.0354
1	0.0272	0.0557	0.0544	0.0694	0.0838	0.0870	0.0944	0.0755	0.0509	0.0646	0.0815	0.0506	0.0459
2	0.0266	0.0544	0.0836	0.0983	0.1124	0.1309	0.1376	0.1222	0.1004	0.0927	0.1072	0.0916	0.0877
3	0.0418	0.0694	0.0983	0.1560	0.1693	0.1902	0.2146	0.1837	0.1634	0.1695	0.1631	0.1240	0.1529
4	0.0410	0.0838	0.1124	0.1693	0.2290	0.2492	0.2762	0.2619	0.2259	0.2343	0.2418	0.1803	0.1870
5	0.0449	0.0870	0.1309	0.1902	0.2492	0.3208	0.3480	0.3305	0.3104	0.3074	0.3174	0.2616	0.2527
6	0.0483	0.0944	0.1376	0.2146	0.2762	0.3480	0.4323	0.4093	0.3853	0.4024	0.4012	0.3398	0.3433
7	0.0266	0.0755	0.1222	0.1837	0.2619	0.3305	0.4093	0.4676	0.4409	0.4384	0.4565	0.4141	0.3869
8	0.0237	0.0509	0.1004	0.1634	0.2259	0.3104	0.3853	0.4409	0.4973	0.4906	0.4890	0.4686	0.4582
9	0.0399	0.0646	0.0927	0.1695	0.2343	0.3074	0.4024	0.4384	0.4906	0.5848	0.5799	0.5104	0.5463
10	0.0406	0.0815	0.1072	0.1631	0.2418	0.3174	0.4012	0.4565	0.4890	0.5799	0.6802	0.6026	0.5934
11	0.0097	0.0506	0.0916	0.1240	0.1803	0.2616	0.3398	0.4141	0.4686	0.5104	0.6026	0.6859	0.6234
12	0.0354	0.0459	0.0877	0.1529	0.1870	0.2527	0.3433	0.3869	0.4582	0.5463	0.5934	0.6234	0.7654

However, as function 2 doesn't give us a hump shape and might not be as accurate as function 7, we abandon this attempt and do the function 7 instead.

3.2 Monte Carlo Simulation with long jump method

3.2.1 Overview

After optimizing the volatility and correlation parameters, we then carried out a Monte-Carlo simulation using the long jump method. Forward rates were simulated using:

$$F_k(T_i) = F_k(T_{i-1})exp\left(\int_{T_{i-1}}^{T_i} \mu_k(F,t) - \frac{1}{2}C_{kk} + \sum_{i=1}^{m} \sigma_{ki}\phi_i\right)$$

The drift term $\mu_k(F,t)$ was calculated iteratively using a predictor corrector method. Using a predictive corrector method allowed us to consider the fact that the drift term was dependent upon the path taken by the forward rates. We assumed that if we chose the longest dated zero-coupon bond $P(t,t_M)$ as the numeraire, the forward rate $F_M(t)$ is driftless and its value at the next time instance can calculated. The drift term can be written in a summation form. The forward rate was then simulated using the following expression:

$$F_k(T_i) = F_k(T_{i-1})exp\left(-\frac{1}{2}\sum_{j=k+1}^{M}\left(\frac{F_j(T_{j-1})\tau_j}{1+F_j(T_{j-1})\tau_j} + \frac{F_j(T_{j-1})\tau_j}{1+F_j(T_{j-1})\tau_j}\right)C_{jk} - \frac{1}{2}C_{kk} + \sum_{i=1}^{m}\sigma_{ki}\phi_i\right)$$

where $\phi_i \approx N(0,1)$ from a univariate normal distribution. C_{jk} and σ_{ki} come from the covariance matrix and the square root of covariance matrix given by the following formula:

$$C_{jk} = \int_{T_{i-1}}^{T_i} \rho_{jk}\sigma_j(t)\sigma_k(t)\,dt$$

3.2.2 problems encountered in simulation

One thing that needs to be specified is that as we move forward in time, our forward rates dead, which results a shrinking in the covariance matrix. At the very beginning, we didn't realize the changing of the covariance matrix. By using one covariance matrix but don't changing the subscript of the correlation and the function correspondingly, we are taking the wrong covariance from the matrix, resulting the forward rate becomes bigger and bigger when moving forward in time.

The error can be illustrated clearly by taking the first term in the matrix as example. We get the forward rate at T_0 by integral the forward rate from now, and the first term in the covariance matrix is calculated by: $C_{11} = \phi_1\phi_1\rho_{11}\int_0^{T_0}\varphi(T_0-t)\varphi(T_0-t)dt$. As we move to T_1 the first term in the covariance matrix then changes to $C_{22} = \phi_2\phi_2\rho_{22}\int_{T_0}^{T_1}\varphi(T_1-t)\varphi(T_1-t)dt$. But we wrongly calculate the term and didn't update the covariance matrix when the time moves forward. After fixing the problem, we get our simulated forward rate correctly.

3.3 Valuation

3.3.1 European Swaption

(1) Product description

Constrained by the availability of the quotes and the maturity of the UBS product, we chose to price ATM swaptions only with maturities + tenors less than or equal to the maturity of the UBS product, that is, three and a half years. Quotes of those ATM Swaptions from Bloomberg are listed as follows.

Maturity/Tenor	1Yr	2Yr	3Yr
3Mo	43.15	41.77	52.7
6Mo	38.7	47.4	48.64
9Mo	45.55	46.24	
1Yr	44.32	44.96	

(2) Valuation procedure

We take the long jump until the maturity date of the swaption $\mathrm{T_0}$, where $\mathrm{T_M}$ is the maturity of the underlying swaption. We need to simulate M forward rates: $F_1(T_0), F_2(T_0), F_3(T_0), \ldots, F_M(T_0)$.

Since $\mathrm{Swaption}(0, \mathrm{T_0}, \mathrm{T_M}) = \max\{\mathrm{Swap\ Value}(0, \mathrm{T_0}, \mathrm{T_M}), 0\}$, set $\mathrm{P}(\mathrm{t}, \mathrm{T_M})$ as the numeraire and we get

$$\frac{\mathrm{Swap\ Value}(0, \mathrm{T_0}, \mathrm{T_M})}{\mathrm{P}(0, \mathrm{T_M})} = \frac{\mathrm{Swap\ Value}(\mathrm{T_0}, \mathrm{T_0}, \mathrm{T_M})}{\mathrm{P}(\mathrm{T_0}, \mathrm{T_M})}$$

Consider $\mathrm{Swap}(\mathrm{T_0}, \mathrm{T_0}, \mathrm{T_M})$ as the price of a coupon payment bond with a coupon rate of K and payment period from $\mathrm{T_1}$ to T_M, the payoff of the swaption at maturity is given by:

$$\mathrm{Swap\ Value}(\mathrm{T_0}, \mathrm{T_0}, \mathrm{T_M}) = 1 - \mathrm{P}(\mathrm{T_0}, \mathrm{T_M}) - \sum_{\mathrm{i}=1}^{\mathrm{M}} \mathrm{K} * \tau_\mathrm{i} * \mathrm{P}(\mathrm{T_0}, \mathrm{T_i})$$

Rearrange the two equations above and we get the discounted payoff:

$$\mathrm{Swap\ Value}(0, \mathrm{T_0}, \mathrm{T_M}) = \frac{1 - \mathrm{P}(\mathrm{T_0}, \mathrm{T_M}) - \sum_{\mathrm{i}=1}^{\mathrm{M}} \mathrm{K} * \tau_\mathrm{i} * \mathrm{P}(\mathrm{T_0}, \mathrm{T_i})}{\mathrm{P}(\mathrm{T_0}, \mathrm{T_M})} * \mathrm{P}(0, \mathrm{T_M})$$

For ATM swaptions, $K = Swap\ rate_{\mathrm{T_0}, \mathrm{T_M}}(0) = \frac{P(0,T_0) - P(0,T_M)}{\sum_{i=0}^{M-1} \tau_i P(0, T_{i+1})}$, and $P(T_0, T_i) = \prod_{k=0}^{i-1} \frac{1}{1 + F(T_0, T_k, T_{k+1}) *}$. By simulating N times, we get N prices for each swaption, The final pricing of each swaption is the average of those N prices.

$$\mathrm{Swaption}(0, \mathrm{T_0}, \mathrm{T_M}) = \frac{1}{N} \sum_{p=1}^{N} \mathrm{Swaption}^p(0, \mathrm{T_0}, \mathrm{T_M})$$

(3) Result

The table below shows the European Swaption price obtained from different model.

Maturity	Tenor	Volatility (Quote/100)	Black price	BDT price	LMM Price
0.25	1	0.4315	0.0014183	0.0011287	0.000268823
0.25	2	0.4177	0.0025993	0.0025289	0.000299401

0.25	3	0.527	0.0047929	0.003939	0.000356782
0.5	1	0.387	0.0017062	0.0015672	0.000392875
0.5	2	0.474	0.0040237	0.0034633	0.000534351
0.5	3	0.4864	0.0060999	0.0053954	0.000630189
0.75	1	0.4555	0.0023773	0.002069	0.000538777
0.75	2	0.4624	0.0047188	0.0044049	0.000773273
1	1	0.4432	0.0025915	0.0025738	0.000628502
1	2	0.4496	0.0052349	0.0053077	0.000842809

Our LMM model tends to have a very strange result on the European swaptions, which is not supposed to be the right result. As LMM model calibrate both to the caplet volatilities and swaptions volatilities, it considered the correlation between forward rates, it should be more accurate in valuing the swaptions. The flaw might be due to the following reasons:

- We choose the simple correlation function that only includes β when calibrating the swaption volatilities. The function might not give us an accurate covariance matrix. Although we tried to improve the correlation function by using three parameters, it failed to converge and didn't give us a result.
- When taking the square root of the covariance matrix, we are supposed to use the Cholesky method. But it went wrong and showed that the array is not positive definite, which might be due to the inaccurate covariance matrix generated from the one parameter correlation function. Thus, we use the normal square root of the covariance matrix. However, as a result, this gave us complex number of the square root, causing our forward rate to be complex number. We have to take the real part of the forward rate in our valuation. This might largely affect the result of our valuation.

3.3.2 Product

In this case, we valued product 3 with LMM model. The product overall features have been introduced in BDT section. The main idea is that, since the product has call feature, we need to calculate the call value and continuation value at each appropriate date, then use Longstaff and Schwartz method to determine which value should be taken. We will start the valuation from the maturity date and move backward, and we move the all the payoffs to the maturity of the numeraire and sum them all and then discounted to time 0.

As we know that on the 29th day of each October and April, there will be a semi-annual payment ($T_i, i\ = 1,3,5,7,9,11$) while on the redeemable date the redemption price equals to the 100% principle plus any accrued and unpaid interest.

(1) The general algorithm

For each Monte Carlo simulation path, we simulate a forward rates matrix and store it. The following forward matrix is a simulation example in our code.

	t=0	t=T0	t=T1	t=T2	t=T3	t=T4	t=T5	t=T6	t=T7	t=T8	t=T9	t=T10	t=T11	t=T12
F1(t)	0.0182	0.0202	-	-	-	-	-	-	-	-	-	-	-	-
F2(t)	0.0165	0.0139	0.0120	-	-	-	-	-	-	-	-	-	-	-
F3(t)	0.0158	0.0169	0.0177	0.0195	-	-	-	-	-	-	-	-	-	-
F4(t)	0.0155	0.0149	0.0168	0.0132	0.0115	-	-	-	-	-	-	-	-	-
F5(t)	0.0153	0.0143	0.0117	0.0128	0.0116	0.0134	-	-	-	-	-	-	-	-
F6(t)	0.0148	0.0160	0.0181	0.0185	0.0183	0.0192	0.0181	-	-	-	-	-	-	-
F7(t)	0.0143	0.0135	0.0151	0.0130	0.0114	0.0155	0.0118	0.0164	-	-	-	-	-	-
F8(t)	0.0153	0.0141	0.0170	0.0141	0.0114	0.0133	0.0148	0.0145	0.0116	-	-	-	-	-
F9(t)	0.0151	0.0148	0.0143	0.0146	0.0134	0.0143	0.0186	0.0173	0.0154	0.0147	-	-	-	-
F10(t)	0.0149	0.0155	0.0133	0.0118	0.0132	0.0126	0.0107	0.0092	0.0114	0.0114	0.0134	-	-	-
F11(t)	0.0147	0.0152	0.0163	0.0176	0.0194	0.0188	0.0236	0.0185	0.0165	0.0175	0.0203	0.0175	-	-
F12(t)	0.0153	0.0161	0.0156	0.0172	0.0206	0.0206	0.0165	0.0154	0.0162	0.0157	0.0159	0.0178	0.0146	-
F13(t)	0.0152	0.0151	0.0156	0.0170	0.0208	0.0203	0.0201	0.0236	0.0225	0.0183	0.0153	0.0158	0.0152	0.0143

Then we chose $P(\mathrm{t}, \mathrm{T}_{13})$ as numeraire. The value of $P(0, \mathrm{T}_{13})$ is known from initial data and we know that $P(\mathrm{T}_{13}, \mathrm{T}_{13}) = 1$. The general formula for other numeraires is:

$$P(\mathrm{T}_i, \mathrm{T}_{13}) = \frac{1}{\prod_{j=i+1}^{13}(1 + F_j(T_i)\tau)}$$

Q_i: the cash flow discounted to T_{13}

At T_{13} (04/29/2023):

$$Q_{13} = 1000 \times \left(1 + 2.5\% \times \frac{180}{360}\right) = 1012.5$$

At T_{12} (01/29/2023):

$$Call\ value = 1000 \times (1 + 2.5\% \times \tfrac{90}{360}) = 1006.25$$

$$Continuation\ value = P(\mathrm{T}_{12}, \mathrm{T}_{13}) \times Q_{13}$$

$$Q_{12} = \begin{cases} \dfrac{call\ value}{P(\mathrm{T}_{12}, \mathrm{T}_{13})}, if\ call\ value < \widehat{cont.value} \\ \dfrac{continuation\ value}{P(\mathrm{T}_{12}, \mathrm{T}_{13})}, otherwise \end{cases}$$

At T_{11} (10/29/2022):

$$Call\ value = 1000 \times (1 + 2.5\% \times \frac{180}{360}) = 1012.5$$

$$Continuation\ value = P(\mathrm{T}_{11}, \mathrm{T}_{13}) \times Q_{12} + 12.5$$

$$Q_{11} = \begin{cases} \dfrac{call\ value}{P(\mathrm{T}_{11}, \mathrm{T}_{13})}, if\ call\ value < \widehat{cont.value} \\ \dfrac{continuation\ value}{P(\mathrm{T}_{11}, \mathrm{T}_{13})}, otherwise \end{cases}$$

Calculate the Q_i all the way to T_0. Particularly, the interest rate changes at T_9.

At T_0 (01/29/2019):

$$Call\ value = 1000 \times (1 + 2.2\% \times \frac{90}{360}) = 1005.5$$

$$Continuation\ value = P(\mathrm{T}_0, \mathrm{T}_{13}) \times Q_1$$

$$Q_0 = \begin{cases} \dfrac{call\ value}{P(\mathrm{T}_0, \mathrm{T}_{13})}, if\ call\ value < \widehat{cont.value} \\ \dfrac{continuation\ value}{P(\mathrm{T}_0, \mathrm{T}_{13})}, otherwise \end{cases}$$

Repeat the above in N times Monte Carlo method and then we get the average price:

$$V = \frac{1}{N} \sum_{p=1}^{N} Q_0^p \times P(0, \mathrm{T}_{13})$$

(2) Longstaff and Schwartz method

In each path, we use Longstaff and Schwartz method to estimate the expected value ($\widehat{cont.value}$).

At T_k, we do the regression with inputs of forward rates and swap rates to get the intercept and coefficients. Then use them to calculate ($\widehat{cont.value}$).

$$\widehat{cont.value} = \widehat{b_0} + \widehat{b_1}x_1 + \widehat{b_2}x_2 + \widehat{b_3}{x_1}^2 + \widehat{b_4}{x_2}^2 + \widehat{b_5}x_1x_2$$

$$where\ x_1 = F_{k+1}(T_k), x_2 = S_{k,13}(T_k)$$

(3) Result

After running 1000 times Monte Carlo, the product price we got was $999.96. Given that UBS AG priced its product at $1000, our estimation is 0.039% away from the true value.

The error of LMM model is smaller than that of BDT model. The reason for a better result of the LMM model is as following:

- LMM model both calibrate to the caplet volatilities and swaption volatilities. It gives a more accurate correlation relationship between forward rates by using the correlation function, comparing to the perfectly correlated assumption in BDT model.
- Compared to piecewise constant volatility in the BDT model, in LMM model we choose function 7 to calibrate and simulate our volatility. The parametric form ensures that volatilities of each forward rates are time homogeneous and generate the typical hump shape. It also makes sure that all the caplet prices exactly matched.

3.3.3 Bermudan Swaption

Here, we choose the same product as in the BDT model. We value a 2-year Bermudan swaption on a 3-year swap issued at time 0 and paying LIBOR while receiving 1.34%. The swaption could be exercised quarterly at times 0.25, 0.50, 0.75, 1.00, 1.25, 1.50, and 1.75. And the notional amount is equal to $1000.

(1) Simulation

Firstly, we simulate the forward rate at all early exercise dates, from T_0 to T_7. The timeline is defined by the following table.

T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7
0.25	0.5	0.75	1	1.25	1.5	1.75	2

On each path, we simulate the following forward rates:

t=0	t= T_0	t= T_1		t= T_7
$F_1^p(0)$	$F_1^p(T_0)$	-	-	-
$F_2^p(0)$	$F_2^p(T_0)$	$F_2^p(T_1)$	-	-
$F_3^p(0)$	$F_3^p(T_0)$	$F_3^p(T_1)$	…	-
…	…	…	…	$F_8^p(T_7)$
…	…	…	…	
$F_{11}^p(0)$	$F_{11}^p(T_0)$	$F_{11}^p(T_1)$	…	$F_{11}^p(T_7)$

(2) Valuation

Then we do the valuation from the maturity date back to today.

At time T_7: on each path, we calculate the payoff of the swaption by:

$$V^p(T_7) = \max\left(\sum_{j=8}^{11} N\tau_j(F_j^p(T_7) - K)P^p(T_7, T_j), 0\right)$$

Where $P^p(T_7, T_j) = \frac{1}{\prod_{k=8}^{j}(1+F_k^p(T_j)\tau_k)}$. We move this value to the maturity of the swap by dividing the discount factor.

$$Q^p(T_7) = \frac{V^p(T_7)}{P^p(T_7, T_{11})}$$

At time T_6 back to T_0 (generate by T_n below), we first calculate the Early Exercise value by:

$$EE^p(T_n) = \sum_{j=n+1}^{11} N\tau_j(F_j^p(T_n) - K)P^p(T_n, T_j)$$

$$P^p(T_n, T_j) = \frac{1}{\prod_{k=n+1}^{j}(1 + F_k^p(T_j)\tau_k)}$$

Then we calculate the discounted continuous value: $CV^p(T_n) = Q^p(T_{n+1}) \times P^p(T_n, T_{11})$. We run an OLS regression between the discounted continuous value and the rates on all the path to get the estimated continuous value. The regression is carried out by:

$$CV^p(T_n) = b_0 + b_1 x_1^p + b_2 x_2^p + b_3 (x_1^p)^2 + b_4 (x_2^p)^2 + b_5 x_1^p x_1^p + \varepsilon^p$$

where $x_1^p = F^p(T_j, T_{j+1})$, $x_2^p = SR^p(T_n, T_{11})$.

After we get the estimated parameters $\widehat{b_0}, \widehat{b_1}, \widehat{b_2}, \widehat{b_3}, \widehat{b_4}, \widehat{b_5}$, we can calculate our estimated continuous value:

$$\widehat{CV^p(T_n)} = \widehat{b_0} + \widehat{b_1}x_1^p + \widehat{b_2}x_2^p + \widehat{b_3}(x_1^p)^2 + \widehat{b_4}(x_2^p)^2 + \widehat{b_5}x_1^p x_1^p$$

Then our Q value on each early exercise date is given by:

$$Q^p(T_n) = \begin{cases} \dfrac{EE^p(T_n)}{P^p(T_n, T_{11})} & if\ EE^p(T_n) > \widehat{CV^p(T_n)} \\ \dfrac{CV^p(T_n)}{P^p(T_n, T_{11})} & otherwise \end{cases}$$

Finally, the value of the Bermudan swaption is given by:

$$V_0 = \frac{1}{N_p}\sum_{p=1}^{N_p} P(0, T_{11})Q^p(T_0)$$

(3) Result

After 1000 times Monte Carlo simulation, we get our swaption price as $4.574. The result is quite close to that of the BDT model. But as we stated above in the European swaption part, there are some concerns in the covariance matrix, thus might cause our LMM model not as accurate as expected.

1. References

https://www.maths.ox.ac.uk/system/files/attachments/M5_Implementation%20of%20the%20LIBOR%20market%20model.pdf

https://fincad.com/resources/resource-library/wiki/what-libor-market-model-lmm

https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/bermuda-swaption/