COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

IEOR E4735 Structured and Hybrid Products

Final Project

Pricing a Quanto Option

Chandana Dayapule(cd3037), Pranathi Pothireddy(pp2665), Aditi Mendiratta(am5007)

supervised by
Dr. Alireza Javaheri

December 2019

Contents

1 Introduction

The project focuses on pricing the cross-currency contract with stock in one currency (EUR) and the settlement of the contract in the payment currency (USD). The below mentioned equation shows the payoff at the maturity i.e 1 year. Stock and the Libor are assumed to have stochastic nature.

In this work, we investigate the pricing, calibration and modeling the cross currecy option.

$$Max\left[0, \left(\frac{S(T)}{S(0)} - k\right)\left(\frac{L(T-\Delta, T-\Delta, T)}{L(0, T-\Delta, T)} - k'\right)\right] \tag{1}$$

where $S(t)$ is the STOXX50E spot price quantoed from EUR into USD $\forall t \in [0, T]$,L(T-$\Delta, T-\Delta, T$) is the 3 month USD LIBOR rate between $T-\Delta$ and T (Δ = 3 months), k and k' are relative strike prices and T is the expiration time.

In section 2 we list out the methodology used to find the evolution of stock price and short rate models. In the later part of this section we discuss the assumptions and limitations considered for these models. This section is also devoted to the detailed derivation of the quanto stock and the formula for evolution of stock. In the Section 3 shows the input data used for building the models. Section 4 delineates about the calibration and result of the instantaneous forward rate, volatility estimation, short rate model and stock price evolution. Section 5 is dedicated to provide the summary of this model and the results.

2 Pricing Methodology and Assumptions

As per equation 1, we need to calculate stock prices in USD and Libor rates.

2.1 Evolution of Stock Price

2.1.1 Methodology

The risk neutral pricing process is used to for valuation of the stock derivation quoted in EUR into USD. The Arbitrage pricing theory and the Fundamental theorem of asset pricing, allow the computation of option values [Wys08]. In the given problem statement the underlying stock in EUR is quotation that is quantoed into USD. Since the payoff is in USD, we let USD be the numeraire or domestic or base currency. Though the traditional approach to solve such complex securities is computationally intensive and complex, we approach this widely accepted Black-Scholes GBM model for valuating the stock prices. As in this case, the implied volatility surface corresponding to vanilla European FX is neither flat nor constant.

$$dS_t{}^{(3)} = (r_{USD} - r)S_t^{(3)} + \sigma_3 S_t^{(3)} dW_t^{(3)} \tag{2}$$

$$dS_t{}^{(2)} = (r_{USD} - r_{EUR})S_t^{(2)} + \sigma_2 S_t^{(2)} dW_t^{(2)} \tag{3}$$

$$dW_t^{(3)} dW^{(2)} = -\rho_{23} dt \tag{4}$$

where we use a minus sign in front of the correlation. The actual underlying becomes as given below [Hau13]:

$$S_t^{(1)} = \frac{S_t^{(3)}}{S_t^{(2)}} \tag{5}$$

Using Ito's we obtain,

$$d\frac{1}{S_t^{(2)}} = -\frac{1}{(S^{(2)})^2} dS_t^{(2)} + \frac{1}{2}.2.\frac{1}{(S_t^{(2)})^3}(dS_t^{(2)})^2 \tag{6}$$

and hence,

$$\begin{aligned} dS_t^{(1)} &= \frac{1}{S_t^{(2)}} dS_t^{(2)} + S_t^{(3)} d\frac{1}{S_t^{(2)}} + dS_t^{(3)} d\frac{1}{S_t^2} \\ &= \frac{S_t^{(3)}}{S_t^{(2)}}(r_{USD} - r)dt + \frac{S_t^{(3)}}{S_t^{(2)}}\sigma_3 + \frac{S_t^{(3)}}{S_t^{(2)}}(r_{EUR} - r_{USD} + \sigma_2^2)dt - \frac{S_t^{(3)}}{S_t^{(2)}}\sigma_2 dW_t^{(2)} + \frac{S_t^{(3)}}{S_t^{(2)}}\rho_{23}\sigma_2\sigma_3 dt \\ &= (r_{EUR} - r + \sigma_2^2 + \rho_{23}\sigma_2\sigma_3)S_t^{(1)} dt + S_t^{(2)}(\sigma_3 dW_t^{(3)} - \sigma_2 dW_t^{(2)}) \end{aligned} \tag{7}$$

Since $S_t^{(1)}$ is a geometric Brownian motion with volatility σ_1, we introduce a new Brownian motion $W_t^{(1)}$ and find

$$dS_t^{(1)} = (r_{EUR} - r + \sigma_2^2 + \rho_{23}\sigma_2\sigma_3)S_t^{(1)} dt + \sigma_1 S_t^{(1)} dW_t^{(1)} \tag{8}$$

$$\begin{aligned} \sigma_3^2 &= \sigma_1^2 + \sigma_2^2 - 2\rho_{12}\sigma_1\sigma_2, \\ \sigma_1^2 &= \sigma_2^2 + \sigma_3^2 - 2\rho_{23}\sigma_2\sigma_3, \\ \sigma_2^2 + \rho_{23}\sigma_2\sigma_3 &= \rho_{12}\sigma_1\sigma_2 \end{aligned} \tag{9}$$

By inserting the above equations we get,

$$dS_t^{(1)} = (r_{EUR} - r - \rho_{12}\sigma_1\sigma_2)S_t^{(1)} dt + \sigma_1 S_t^{(1)} dW_t^{(1)} \tag{10}$$

This is the Risk Neutral Pricing process that is used for the valuation of the complex option on $S_t^{(1)}$ which is quantoed into USD.

2.1.2 Assumptions and Limitations

The model assumed for the Stock price is a Black Scholes model which has the following assumptions and limitations:

1. Underlying stock prices are log normally distributed
2. Underlying stocks does not pay any dividends
3. Stocks have constant volatility
4. Options are European and can be exercised only at maturity
5. No transaction fees for buying and selling options
6. Markets are perfectly liquid
7. Short selling is allowed without any restrictions

2.2 Interest Rate Model

Short Rate Model

2.2.1 Methodology

We assume that our short rate follows the dynamics of the Hull-While model as given below [Bjo09a]:

$$dr(t) = (\theta(t) - ar(t))\, dt + \sigma dW(t) \tag{11}$$

where r(t) is the short term rate in USD, 'a' is a constant, σ is constant volatility and $\theta(t)$ is a deterministic function given by equation 19.

We followed these steps to calculate the short rate-

1. *Calculate instantaneous forward rates f(t, T) from Zero Coupon Interest Rates Curve*

   According to an article posted by the Bank for International Settlements(BIS), one of the most popular curve assumption for the zero coupon interest rates(s_T) curve is the Nelson and Siegel Parametric Model[BIS05].
   In this method, we determine the parameters of the instantaneous forward rate function by a minimization problem. We use the currently available Libor interest rates data available to us for different times to

maturity(T) and fit it to the given curve and we can get the values of the constants β_0, β_1, β_2 and τ for time t using the following equations:

$$s_T = \beta_0 + (\beta_1 + \beta_2)\frac{\tau}{T}\left(1 - exp\left(\frac{-T}{\tau}\right)\right) - \beta_2 exp\left(\frac{-T}{\tau}\right) \tag{12}$$

Using equation 12, we can now calculate the instantaneous forward rate since by [Bjo09b] and [BIS05] we have the following relationships:

$$f(t,T) = \beta_0 + \beta_1 exp\left(\frac{-T}{\tau}\right) + \beta_2 \frac{T}{\tau} exp\left(\frac{-T}{\tau}\right) \tag{13}$$

$$p^*(0,T) = -\frac{1}{(1+s_T)^T} \tag{14}$$

where p*(t,T) is a zero coupon bond price implied from the market with par value assumed to be \$1.

2. *Calculate interest rate volatility via Zero Coupon Bonds Price(ZBP)*

   Using the ZBP data available to us, we can use the analogy that the price ZBP(t,T,S,X) at time t is the price of a European put option with strike X and maturity T discounted at time S.

$$ZBP(t,T,S,X) = XP(t,T)\Phi(-h + \sigma_p) - P(t,S)\Phi(-h) \tag{15}$$

   where h and σ_p given by:

$$h = \frac{1}{\sigma_p} ln \frac{P(t,S)}{P(t,T)X} + \frac{\sigma_p}{2} \tag{16}$$

$$\sigma_p = \sigma\sqrt{\frac{1 - e^{-2a(T-t)}}{2a}} B(T,S) \tag{17}$$

   and

$$B(t,T) = \frac{1}{a}\left(1 - e^{-a(T-t)}\right) \tag{18}$$

   The above mentioned relation is used in the optimization problem to minimize the error between ZBP and the volatility function to get an optimal value of σ.

3. *Calculate $\theta(t)$ using volatility and instantaneous forward rates estimations*

   From the above two steps, we get σ and f(t,T). We can substitute these values for a given value of T and determine the value of $\theta(t)$ which is given by [Bjo09a]:

$$\theta(t) = \frac{\partial f(0,t)}{\partial T} + af(0,t) + \frac{\sigma^2}{2a}\left(1 - e^{-2at}\right) \tag{19}$$

where

$$\frac{\partial f(t,T)}{\partial T} = \frac{1}{\tau} exp\left(\frac{-T}{\tau}\right)\left(-\beta_1 + \beta_2\left(1 - \frac{T}{\tau}\right)\right) \tag{20}$$

4. *Calculate short rate r(t)*

   Using equations 11, 17 and 19, we have all the data required to calculate r(t). We can calibrate the stochastic process equation using Monte Carlo techniques presented in section 4.

5. *Calculate bond price p(t,T) from hull white term structure equation*

   The hull white equation to find the zero coupon bond prices is as follows [Bjo09a]:

$$p(t,T) = \frac{p^*(0,T)}{p^*(0,t)} exp\left(B(t,T)f^*(0,t) - \frac{\sigma^2}{4a}B^2(t,T)(1 - e^{-2at}) - B(t,T)r(t)\right) \tag{21}$$

   where $f^*(0,t)$ is the market implied instantaneous forward rate given by equation 13 and B(t,T) is given by equation 18.

2.2.2 Assumptions and Limitations

1. 'a' is assumed as 0.03 (as per class discussions).
2. Zero coupon bond price (p(t,T)) par value is assumed to be \$1.
3. Used minimize function from scipy.optimize in the optimization of parameters of forward rate model and stock volatility. This heavily depends on the initial parameter assumptions.
4. Hull-White model assumed for the short rate model has a few limitations[MS]:
   (a) It allows negative rates, which might give an error in the pricing of non linear instruments.
   (b) The price is likely to be sensitive to correlations of these rates if the payoffs are defined in terms of multiple rates or the same rate at different times which is not expressed by this model.

(c) The model cannot generate a wide range of volatility surfaces satisfactorily, due to lack of free calibration parameters.

5. Nelson-Siegel model have following limitations which can be reduced by using ridge regression[JA13]:

   (a) Model may not give observed zero yields in stressed market environment.

   (b) Model might show multi-collinearity depending on estimated parameters.

   (c) The estimated parameters behave erratically and have high variance.

3 Market Data Inputs and Sources

For forward rate curve estimations, we used the Libor Interest Rates in USD on 12^{th} December,2019 available online [lib19] as mentioned in table 3.
For the stock price evolution and short rate calibration, we used Bloomberg to get the data as expressed in table 3. To estimate

Symbol	**Value**	**Description**
S(0)	3710	STOXX50E Price
a	0.03	constant for short rate model
X	1.0075	Strike rate in ZBP
ZBP	0.03	Zero Base Pricing (Caplet Price)
r(0)	0.2	Initial short rate estimate (overnight Libor)
σ_1	0.14	STOXX50E at the money volatility
σ_2	0.045	Euro-USD at the money volatility
q	0	Dividends
r	0.14	EURUSD risk free rate
ρ_{23}	0.53	Eur-USD Correlation

Table 1: Data used, source: Bloomberg

Libor Interest Rates	
Time to Maturity	Rates (%)
1 month	1.74%
2 months	1.83%
3 months	1.89%
6 months	1.89%
1 year	1.93%

Table 2: Libor Interest Rates, source: [lib19]

4 Calibration Approaches and Results

4.1 Instantaneous forward rate estimations and curve fitting

As seen in figure 1, the input spot rates(represented by green dots) are used to build the Nelson-Siegel spot rate curves by ordinary least square estimates. The values of the coefficients after the regression are provided in table 3.
We observe:

1. For longer maturities, the spot and forward rate approaches the value of β_0 asymptotically.

2. For shorter maturities(near 0), the forward and spot rates approaches the value $(\beta_0 + \beta_1)$, which we see from our results in table 3 is positive.

3. β_2 is responsible for the magnitude of the hump and since $\tau > 0$, the direction of the hump in the mid-range maturity is upwards.

These characteristics are in line with the general conclusions in the Nelson and Siegel model and since this method is supported by various central banks[BIS05], we can thus justify the use of this model in our estimations of the hull-white short rate.

Coefficient	**Value**
β_0	0.01821
β_1	-0.00179
β_2	0.00609
τ	0.24966

Table 3: Results for Libor spot rates fitting

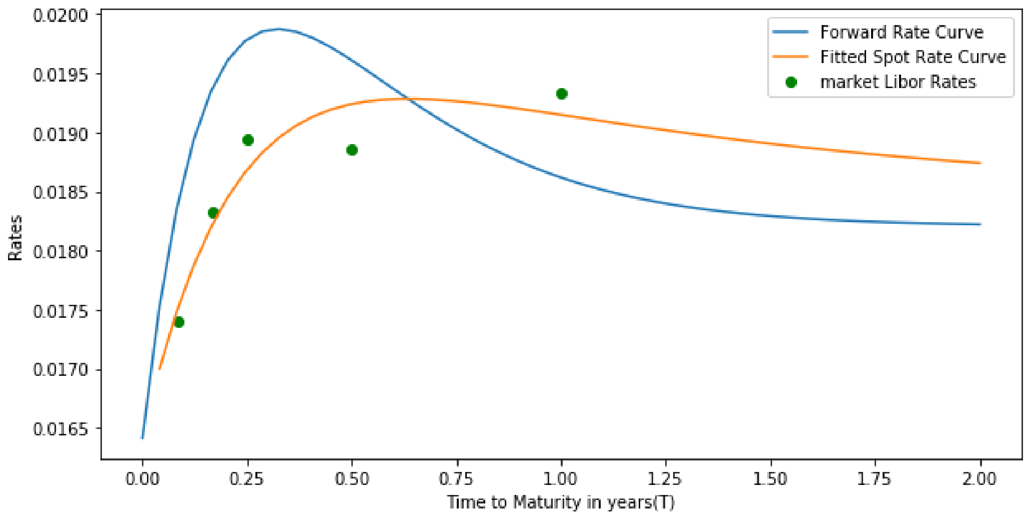

Figure 1: Fitted spot rates by Nelson and Siegel method and corresponding forward rates as per section 2.1.

4.2 Interest rate volatility(σ) estimations and curve fitting

From figure 2, we observe that the interest rate volatility estimates as per section 2.1 gives higher estimates for a shorter time to maturity at the present time(t=0).For longer time to maturity, it approaches zero.
Practically, as the time to maturity becomes larger, the prediction of interest rates becomes less accurate as compared to that of shorter time, hence the volatility should rather increase than approaching zero. This indicates that the volatility fitting model that we used in section 2.2 is applicable only for short rate models and might not give stable results as the time to maturity increases.
Also, this model is highly sensitive to the initial feasible value that we use in the minimization problem. This is due to the optimiser limitations as mentioned in section 2.2.2.

4.3 Short Rate Calibration

The Hull-White process is constructed by passing the term-structure, a and sigma which is used from the above discussed data.

We approached the model my simulating the path generator random sequence along with other simulation inputs as the timestep as one day and the initial short rate value as 0.2 percent as mentioned in the table.

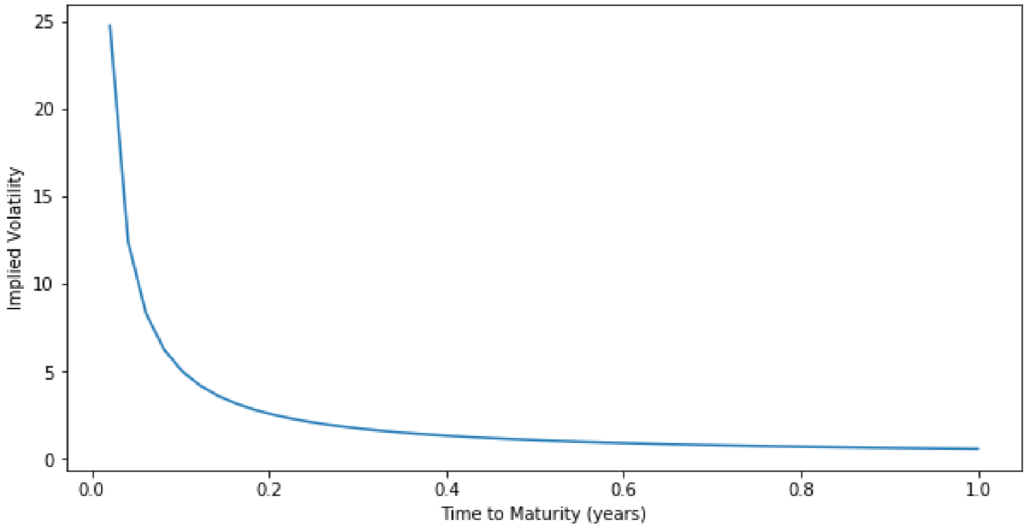

Figure 2: Fitted volatility function(σ) as per section 2.2.

The constants that we used for this model is all defined as shown in the table. Variables sigma and a are the constants that define the Hull-White model. In the simulation, we discretize the time span of length 1 year into 360 intervals (one day) as defined by the timestep variable.

Apart from showing the evolution of short rates we have further plotted the 5 percent to 95 percent quantile and histogram plots for understanding the convergence of the Hull-White model. These results are mentioned in the graphs below we have further made analysis on varying the input parameters like sigma and a and their dependency on the convergence of the model.

4.4 Stock Price Evolution

We used Monte Carlo simulation methodology for the behavior of stock prices which follows a random walk. The first term is equivalent to the drift and the second term is the shock term. To illustrate we have performed the simulations for the daily time step and repeated the simulations for thousand simulations. Few key inputs were obtained from Bloomberg the details of these are mentioned in the input data section. The day zero stock price is taken as 3170 dollars which is obtained from the stoxx data which is mentioned in the table mentioned. The simulations that were performed gave us future outcomes which were used to calculate the price of the exotic option. 10.

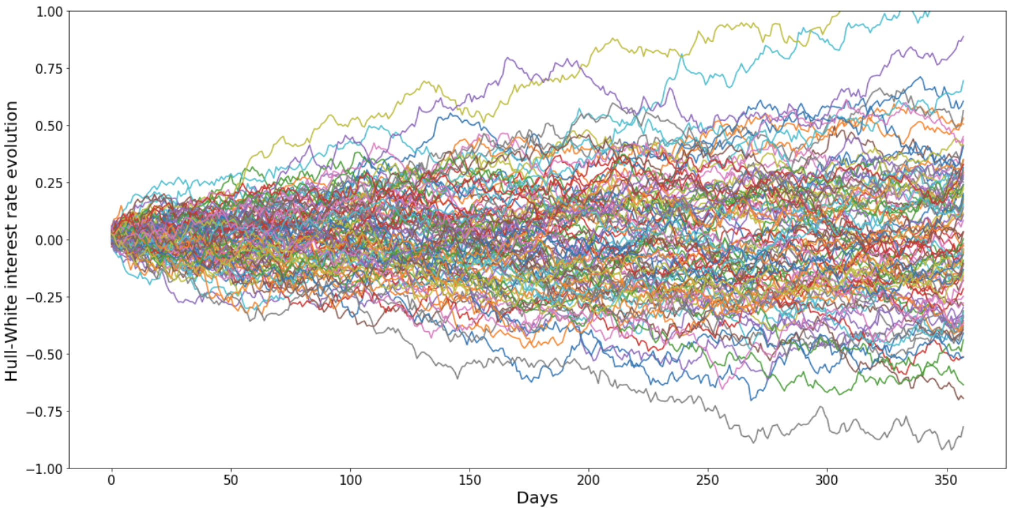

Figure 3: Calibrated Hull-White short rate r(t) as per section 2.2

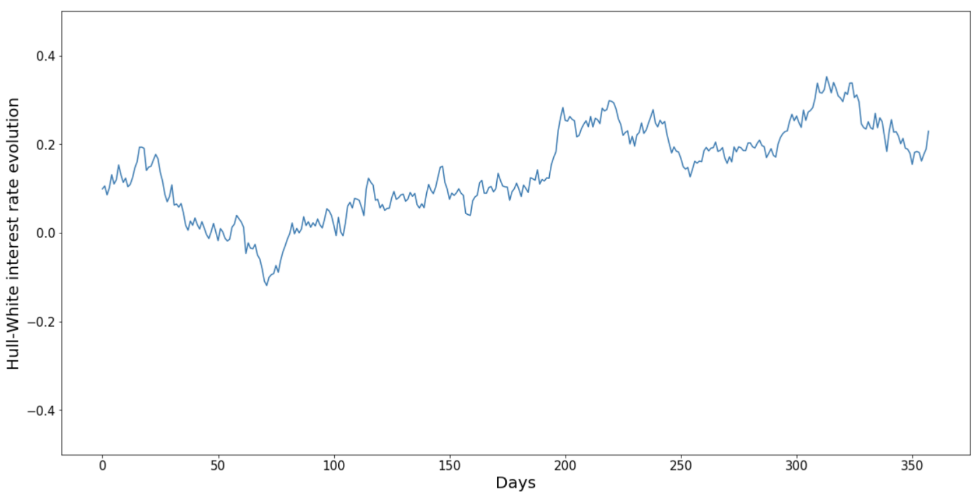

Figure 4: Calibrated Hull-White short rate r(t) for a single iteration as per section 2.2

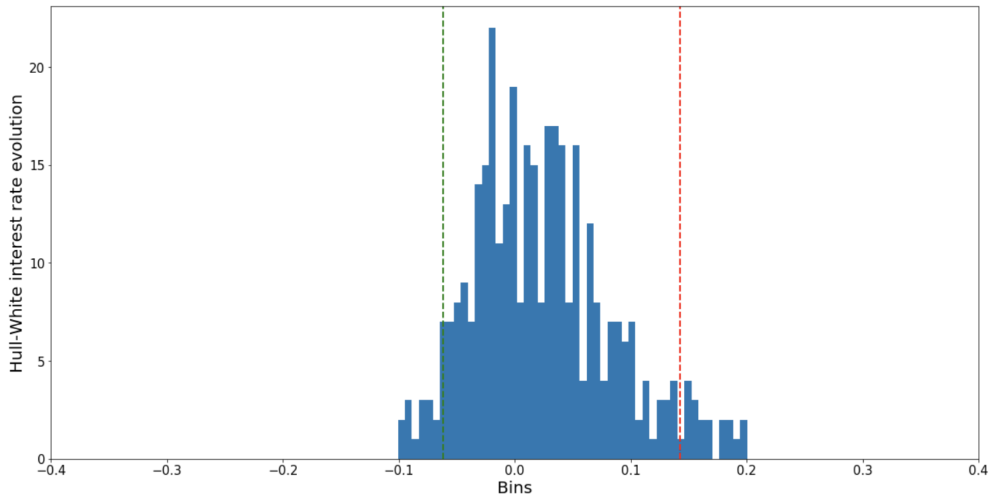

Figure 5: Short rate estimation histogram with 5% and 95% confidence markers.

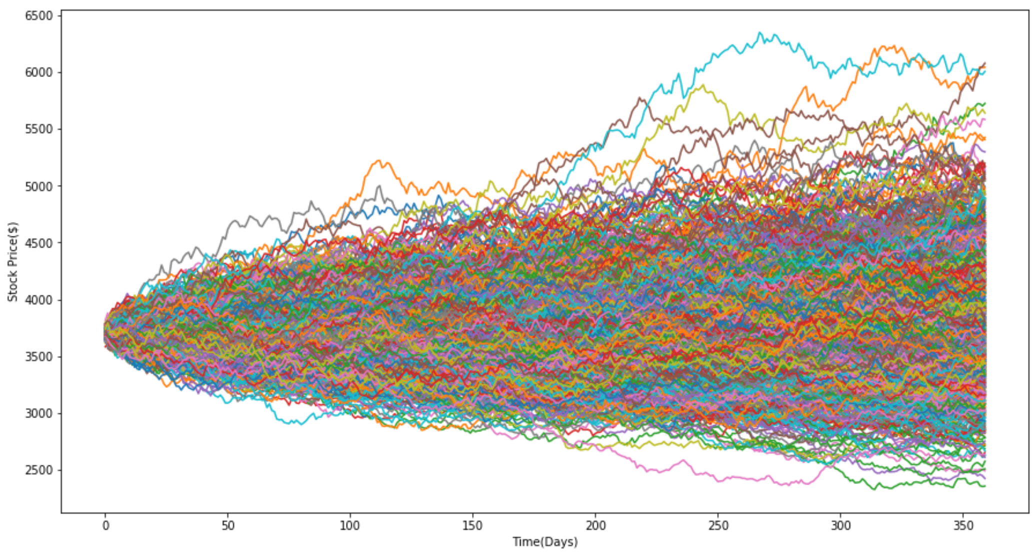

Figure 6: Calibrated stock price as per section 2.1.1.

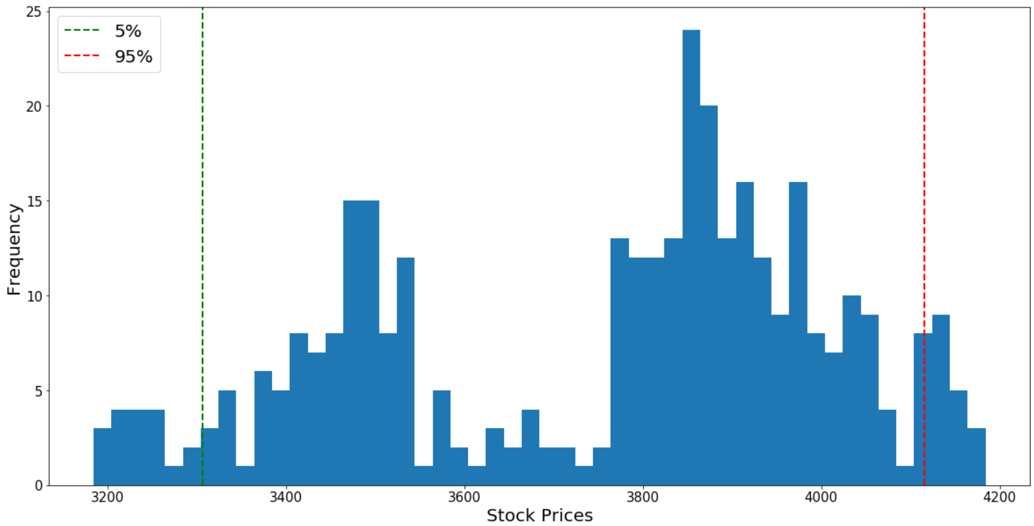

Figure 7: Stock Price estimation histogram with 5% and 95% confidence markers.

5 Conclusion

By performing the libor and the stock simulations and finding the discounted values of the payoff generated at the end of 1 year we obtained the price of the exotic option to be in the range of 0.44 with standard deviation +/-20 percent.

We also found that a few estimates like volatility and eventually, the short rate model is very sensitive to the value of parameters and the innate assumptions of the market value. Therefore, we should account for these factors explicitly while presenting the results and during model calibration.

Monte Carlo simulations to calibrate stock evolution and interest rates proved to be effective in converging to a stable option value with a reasonable volatility estimate as per the investor's risk appetite.

Another important takeaway of using the short rate model(Hull-White) is that it was effective in estimating how much the interest rate could change i.e. predicting the variability rather than the path of rates. Hence, it was more suitable for option pricing as compared to other models that focus more on only predicting values, for example - time series models.

For implementation, refer to the python notebook in appendix.

A Implementation

Implementation is done using python 3.6. We have used libraries like numpy and scipy to make the implementation faster and for better memory management.

Stock Estimation

```
def plot_Stocks(s0=3710,r0=2,r=0.014,q=0,rho=0.53,sig1=0.14,sig2=0.045):

    plt.figure(figsize=(20,10))
    plt.xlabel("Time(Days)",fontsize=20)
    plt.xticks(fontsize=15)
    plt.yticks(fontsize=15)
    plt.ylabel("Stock_Prices",fontsize=20)

    for j in range(1,1000):
        x   = []
        k1 =1
        k2 =1
        s = np.zeros(361)
        s[0]=  3710
        dt   = 1/360
        w1   = np.random.normal(0,1,360)
        w2   = np.random.normal(0,1,360)

        sig3= np.sqrt(sig1**2+sig2**2+2*rho*sig1*sig2)

        w3   = (sig1*w1+sig2*w2)/sig3
        for i in range(1,361):
            s[i] = s[i-1]*(np.exp(((r-q+rho*sig1*sig2)*dt)+(sig3*dt**0.5*w3[i-1]))
            x.append(s[i])
        plt.plot(x)
   plt.show()

plot_Stocks()
```

Figure 8: Stock Price estimation code

Forward rate model

```
#Libor rates used as spot rates for forward rate curve fitting
time_array = np.array([1./12,2./12,3./12,6./12,1.0])
libor_rates = np.array([1.73975/100 ,1.83188/100 ,1.89363/100,1.88638/100,1.93288/100])  #in USD

def zcbCurve_obj_func(params,time_array,libor_rates):
    func= params[0] + (params[1]+params[2])*params[3]/time_array * (1 - np.exp(-time_array/params[3])) - params[2]*np.
exp(-time_array/params[3])
    return np.sum((libor_rates-func)**2)
def constraint1(param):
    return param[0]
def constraint2(param):
    return param[0]+param[1]
def constraint3(param):
    return param[3]
def get_zcbCurve_params(time_array,libor_rates):
    initial_guess = np.array([0.,0.,0.,0.25])
    cons1 = {'type':'ineq','fun':constraint1}
    cons2 = {'type':'ineq','fun':constraint2}
    cons3 = {'type':'ineq','fun':constraint3}
    cons = [cons1,cons2,cons3]
    return minimize(zcbCurve_obj_func,initial_guess,args=(time_array,libor_rates), method='SLSQP',constraints=cons)
sol = get_zcbCurve_params(time_array,libor_rates).x
def f_star(input_time, plot_curves=False):
    if plot_curves==True:
        plt.figure(figsize=(10,5))
        plt.plot(time_array,libor_rates, label = 'actual libor rates curve')
        plt.plot(time_array,sol[0] + (sol[1]+sol[2])*sol[3]/time_array * (1 - np.exp(-time_array/sol[3])) - sol[2]*np.
exp(-time_array/sol[3]),label='computed  libor rates curve')
        plt.plot(time_array,sol[0] + (sol[1]*np.exp(-time_array/sol[3])) + sol[2]*time_array/sol[3]*np.exp(-time_array
/sol[3]),label='forward rates')
        plt.xlabel("Time to Maturity(T)")
        plt.ylabel("Rates(%)")
        plt.legend()
        plt.show()
    return sol[0] + (sol[1]*np.exp(-input_time/sol[3])) + sol[2]*input_time/sol[3]*np.exp(-input_time/sol[3])
def compute_df(T):
    return 1/sol[3] * np.exp(-T/sol[3])*(-sol[1] + sol[2] * (1 - T/sol[3]))
def p_star(input_time):
    if input_time==0:
        input_time = 0.000000001 #divide by zero error
    spot_rate = sol[0] + (sol[1]+sol[2])*sol[3]/input_time * (1 - np.exp(-input_time/sol[3])) - sol[2]*np.exp(-input_t
ime/sol[3])
    return 1/(1+spot_rate)**input_time
f_star(1, True)
```

Figure 9: Forward Rate Model Code

Stock Estimation

```
def plot_Stocks(s0=3710,r0=2,r=0.014,q=0,rho=0.53,sig1=0.14,sig2=0.045):

    plt.figure(figsize=(20,10))
    plt.xlabel("Time(Days)",fontsize=20)
    plt.xticks(fontsize=15)
    plt.yticks(fontsize=15)
    plt.ylabel("Stock_Prices",fontsize=20)

    for j in range(1,1000):
        x   = []
        k1 =1
        k2 =1
        s = np.zeros(361)
        s[0]=  3710
        dt  = 1/360
        w1  = np.random.normal(0,1,360)
        w2  = np.random.normal(0,1,360)

        sig3= np.sqrt(sig1**2+sig2**2+2*rho*sig1*sig2)

        w3  = (sig1*w1+sig2*w2)/sig3
        for i in range(1,361):
            s[i] = s[i-1]*(np.exp(((r-q+rho*sig1*sig2)*dt)+(sig3*dt**0.5*w3[i-1]))
            x.append(s[i])
        plt.plot(x)
   plt.show()

plot_Stocks()
```

Figure 10: Stock Price estimation code

Simulating Hull-White model

```
def simulate_r_hull(t):
    w1  = np.random.normal(0,1,361)

    r   = np.zeros(361)
    dt  = 1/360
    for i in range(1,361):
        r[i] = r[i-1]+( get_theta(i/360,T,S)-a*r[i-1])*dt+sig*(dt**.5)*w1[i-1]

#          print(r[i])
    return(r[int(t*359)])

simulate_r_hull(0.2)
```

Figure 11: Hull White Model Code

Simulating Libor rates

```
def L(t,S,T):
    p_T= p(t,T)
    p_S= p(t,S)
    return -(p_T-p_S)/((T-S)*p_T)


(L(0.75,0.75,1)/L(0,0.75,1))-0.1
```

Figure 12: Libor Model

Simulating Price

```
def price_(s0=3710,r0=0.2,r=0.014,q=0,rho=0.53,sig1=0.11,sig2=0.043):
    p=[]
    for j in range(1,1000):

        s   = np.zeros(361)
        k1  =1
        k2  =1

        s[0]= s0
        dt  = 1/360
        w1  = np.random.normal(0,1,360)
        w2  = np.random.normal(0,1,360)

        sig3= np.sqrt(sig1**2+sig2**2+2*rho*sig1*sig2)
        w3  = (sig1*w1+sig2*w2)/sig3

        for i in range(1,361):
            s[i] = s[i-1]*(np.exp(((r-q+rho*sig1*sig2)*dt)+(sig3*dt**0.5*w3[i-1])))
            r[i] = r[i-1]+( get_theta(i/360,T,S)-a*r[i-1])+sig*(dt**.5)*w1[i-1]

        p.append(np.exp(-r[360])*(max((s[360]/s0-1)*(l[j]),0)))

    return np.mean(p)
price_()
```

Figure 13: Price Simulation of Exotic Option

References

[BIS05] BIS. Zero-coupon yield curves estimated by central banks. https://www.bis.org/publ/bppdf/bispap25a.pdf, 2005.

[Bjo09a] Tomas Bjork. *Arbitrage Theory in Continuous Time*, chapter Martingale Models for the Short Rate, pages 383–385. In [Bjo09b], third edition, 2009.

[Bjo09b] Tomas Bjork. *Arbitrage Theory in Continuous Time*. Oxford University Press Inc., third edition, 2009.

[Hau13] Martin Haugh. Foreign exchange, adr's and quanto-securities. 2013.

[JA13] Marc J.K.De Ceuster Hairui Zhang Jan Annaert, Anouk G.P.Claes. Estimating the spot rate curve using the nelson–siegel model: A ridge regression approach. 2013.

[lib19] Usd libor interest rate - us dollar libor. https://www.global-rates.com/interest-rates/libor/american-dollar/american-dollar.aspx, 2019.

[MS] Mario Zacharias Marcus Scheffer. A comparitive study of the 1-factor hull white and the g2++ interest rate model. https://jp.milliman.com/uploadedFiles/insight/2018/A_Comparative_Study_of_the_1-Factor_Hull_White.pdf.

[Wys08] Uwe Wystup. Quanto options. https://mathfinance.com/wp-content/uploads/2017/06/wystup_quanto_eqf.pdf, 2008.