Fixed Income Derivatives Final Exam Fall 2023(24.02.2024) - Solution

Problem 1

Imagine that you are working for a financial institution and that a client has approached you to help him manage his loan obligations. The client has a loan on which he has exactly 6 year remaining and on which he pays 6M EURIBOR semi-annually. That is, the first payment on the loan takes place in exactly six months and the first 6M EURIBOR fixing has just been announced. For simplicity, you assume the notional on the loan is 1 EUR. The client is worried that interest rates might rise in the future and he would therefore like to protect himself against rising interest rates. To be able to help the client, you decide to use the Vasicek model and fit the model to a set of continuously compounded zero coupon spot rates that your colleague has extracted from market data and which are given in the table below. The spot rates are denoted by $R(0,T)$ and the corresponding maturities by T.

T	0.1	0.25	0.5	0.75	1	1.5	2	3	4	5	7	10
$R(0,T)$	0.0334	0.0352	0.0375	0.0392	0.0405	0.0422	0.0433	0.0445	0.0451	0.0455	0.0459	0.0462

In the Vasicek model, the dynamics of the short rate r_t under the risk neutral pricing measure $\mathbb{Q}$ are

$$
\begin{aligned}
dr_t &= (b - ar_t)dt + \sigma dW_t, \quad t > 0 \\
r(t=0) &= r_0
\end{aligned}
\tag{1}
$$

From years of experience, you know that $\sigma = 0.04$ so you will *not* need to estimate σ.

a) Using initial values $\tilde{r}_0 = 0.038$, $\tilde{a} = 1.2$, $\tilde{b} = 0.07$, fit a Vasicek model to the data from your colleague and estimate r_0, a and b.

   i) Report the estimates $\hat{r}_0$, $\hat{a}$ and $\hat{b}$.

   ii) Demonstrate that a Vasicek model with your estimated parameter values fits the data.

b) The first option you have decided to provide the client is that he can switch his 6M EURIBOR floating rate coupon payments into fixed rate coupon payments also paid semi-annually by entering into a 6Y payer swap. The second option you will provide the client is that he can enter into an interest rate cap preventing any of the floating rate payments over the next 6 years from exceeding some strike. So that the client can better compare his options, you decide to set the strike of the interest cap equal to the par swap rate for the 6Y interest rate swap. If, for some reason, you were not able to compute the par swap rate, you can set the strike of the interest rate cap to 0.05.

   i) Find the par swap rate for the proposed interest rate swap.

   ii) Using a Vasicek model and the parameters you have found, compute the price of the interest rate cap both if the client pays upfront or if the client chooses to pay a spread alongside his normal coupon payments. In your computation, you should use that an interest rate cap consists of a number of caplets, that caplets are essentially European options on a zero coupon bond and that such European option prices can be computed explicitly in the Vasicek model.

c) Now, the interest rate swap costs nothing but the interest rate cap comes with a cost. Explain if and how/how not, this makes sense in an arbitrage-free market by briefly discussing the potential upside and downside of the two methods of managing interest rate risk, presented to the client.

Problem 1 - Solution

a) i) Fitting the Vasicek model is relatively trivial and can be done in many ways but the result should be roughly that $\hat{r}_0 = 0.034$, $\hat{a} = 0.8$ and $\hat{b} = 0.042$.

ii) The fit of the model should be very good which can demonstrated by reporting the mean squared error of the fitted parameters, or by plotting the fitted versus observed values of the short rate.

b) i) The par swap rate can be found to be approximately $R_{swap} = 0.048065$.

ii) To compute the price of the interest rate cap, we need to find the price $\Pi(T_{i-1})$ of each of the caplets on the 6M EURIBOR fixings for $T_{i-1} = \{0.5, 1, ..., 3.5\}$. Now, the price of a caplet with strike R can be seen as $\big(1 + \alpha R\big)$ put options with a strike of $\frac{1}{1+\alpha R}$. The caplet prices in bps become

T_{i-1}	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5	5.5
$\Pi(T_{i-1})$	26.056	39.242	45.940	49.326	50.863	51.320	51.129	50.546	49.727	48.767	47.727

The upfront price of the interest rate cap becomes roughly 0.0511 or 511 bps. If paid semiannually, the price of the cap becomes 49.22 bps.

c) The interest rate swap eliminates all uncertainty about fixed future coupon payments. However, this option also has no upside in that the investor will not benefit if future 6M EURIBOR fixings are low. The interest rate cap prevents interest rate payments from becoming very large but retains the upside that the client will benefit from low future EURIBOR fixings. Therefore, there is nothing surprising about the fact that choosing the interest rate cap comes at a cost.

Problem 2

Your job is to assess the risk of your institution's interest rate swap exposure and therefore, you must first calibrate a zero coupon term structures of interest rates to market data. The 6M EURIBOR fixing has just been announced to be 0.04110 and in addition, you have the data shown in the table below for forward rate agreements and interest rate swaps. The interest rate swaps pay 6M EURIBOR semi-annually against the fixed par swap rates listed below also paid semi-annually.

EURIBOR	Fixing	FRA	Midquote	IRS	Midquote
6M	0.04110	1X7	0.04358	2Y	0.03824
		2X8	0.03295	3Y	0.04083
		3X9	0.03418	4Y	0.04242
		4X10	0.03531	5Y	0.04346
		5X11	0.03635	7Y	0.04468
		6X12	0.03731	10Y	0.04561
		7X13	0.03819	15Y	0.04633
		8X14	0.03900	20Y	0.04667
		9X15	0.03975	30Y	0.04700

a) Calibrate a zero coupon term structure of continuously compounded spot interest rates to the 6M EURIBOR fixing, the forward rate agreement rates and par swap rates given in the table above.

i) Report zero coupon spot rates for the following six maturities: $T = [0.5, 1, 3, 5, 10, 20, 30]$ based on your calibration.

ii) Plot the term structures of continuously compounded zero coupon spot rates for maturities ranging from 0 to 30 years.

b) Compute par swap rates for all maturities T where $T \in \{1, 1.5, 2, 2.5, 3, ..., 30\}$

i) Plot the term structure of par swap rates in the same plot as the spot rates.

ii) Are the par swap rates you have calculated very close to the spot rates or are they very far?

iii) Briefly discuss your finding in ii) and perhaps relate your answer to the nature of the stream of cashflows from an interest rate swap and a zero coupon bond.

c) Your colleague entered into a 7Y *receiver* swap receiving a fixed coupon of 0.047 semi-annully against paying 6m EURIBOR semiannually exactly one year ago today. Your colleague would like to know how much he has lost or gained on his position based on your zero coupon yield curve. You can assume that the notional of his swap position was 1 when he entered into the swap.

i) Report the profit or loss of your colleague as a percentage of the notional.

ii) Based on whether your colleague lost money or not, determine whether interest rates were generally higher or lower one year ago.

Problem 2 - Solution

a) Calibration of the continuously compounded zero coupon spot rate curve can be performed but the exact values of the spot rates will depend on the method used in the interpolation.

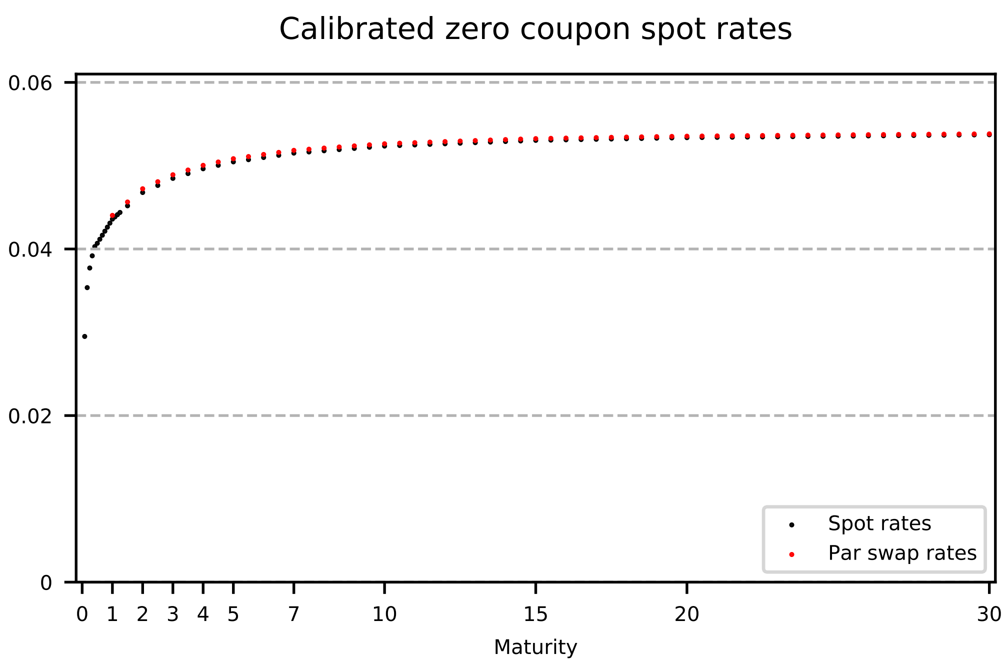
i) Spot rates for the maturities $T = [0.5, 1, 3, 5, 10, 20, 30]$ become roughly

T	0.5	1	3	5	10	20	30
$R(0,T)$	0.04068	0.04359	0.04848	0.05046	0.05236	0.05337	0.05372

ii) The plot of the zero coupon spot rate curve can be seen in b).

b) Computing par swap rates should be done by first computing the accrual factor of the swap.

i) The plot becomes



ii) The swap rates are almost identical to the spot rates.

iii) Zero coupon spot rates and par swap rates for same maturities are almost always very close. The main reason for this result is that a swap and a zero coupon bond on average, over the life of these securities, have very similar cashflows. A zero coupon bond pays no coupons and the *net* coupon to a swap being the difference between a fixed and a floating rate is on average also close to zero. A zero coupon bond pays the whole notional at maturity to the lender, and though there is no exchange of notional in practice, a swap also works on a notional repaid at maturity.

c) i) To solve this problem, note that the colleagues 7Y swap has now become a 6Y swap. The PNL of the colleagues position in the receiver swap can be computed as

$$\text{PNL} = (K - R_{swap}) \cdot S_{swap} \approx -0.02232 \approx -223 \text{ bps}$$

ii) A receiver swap is positioned for falling interest rates and since the colleague lost money on his position, we can only assume that rates have risen since he entered into the swap. The 6Y par swap rate is now 0.05135 and hence, the par swap rate is today higher than the par swap rate of 0.04700 when the colleague entered into the swap. This corresponds to a rate increase of roughly 41 bps which when multiplied by the accrual factor slightly above 5 does indeed yield a little more than 200 bps.

Problem 3

You work for a major financial institution and your colleague has just supplied you with zero coupon bond prices extracted from risk-free traded securities. The prices at present time $t = 0$ for a range of maturities T are shown below.

Table 1: Zero coupon bond prices

T	0.50	1.00	1.50	2.00	2.50	3.00	3.50
$p(0,T)$	0.98314916	0.96478677	0.94539738	0.92535353	0.90493951	0.88437071	0.86380916
T	4.00	4.50	5.00	5.50	6.00	6.50	7.00
$p(0,T)$	0.84337571	0.8231596	0.80322594	0.78362142	0.76437872	0.74551992	0.72705911

You additionally have prices of 2Y5Y payer swaptions, denoted $\Pi_{swaption}$, for a notional of 1 and different strikes. The underlying swap pays a fixed rate semi-annually against 6M EURIBOR received semi-annually and the prices of the swaptions are given in the table below. Recall that a 2Y5Y payer swaption gives you the right but not obligation to enter into a 5 year payer swap at the time of exercise in 2 years. The strikes are to be interpreted such that a K_{offset} of 0 (ATMF) corresponds to a strike equal to the current 2Y5Y forward rate and a K_{offset} of 100 say corresponds to a strike 100 bps above the current 2Y5Y forward rate.

Table 2: $2Y5Y$ Swaption prices

$K_{offset}(bp)$	-300	-250	-200	-150	-100	-50	ATMF
$\Pi_{swaption}$	0.12256859	0.10253932	0.08273803	0.0633625	0.04480655	0.02793572	0.0145331
$K_{offset}(bp)$	50	100	150	200	250	300	-
$\Pi_{swaption}$	0.00650867	0.0030062	0.00158778	0.00094974	0.00062285	0.00043427	-

a) Compute the par swap rate for the underlying asset which in this case is the 2Y *forward* 5Y interest rate swap by first computing it's accrual factor. Then compute the remaining strikes and solve the problems below.

i) Report the $2Y5Y$ forward par swap rate.

ii) Compute Black implied volatilities $\bar{\sigma}_i$ for all strikes and plot these as a function of K_{offset}.

iii) Interpret the implied volatility plot and assess if the market is pricing swaptions according to Black's model. What can be said about the distribution of the $2Y5Y$ forward par swap rate implied by the pricing measure chosen by the market and how does that distribution compare to the log normal distribution?

You now decide to fit a SABR model to the implied volatilities you have computed from market swaption prices. The SABR model you should consider is of the usual form given below

$$
\begin{aligned}
dF_t &= \sigma_t F_t^{\beta} dW_t^{(1)}, \qquad F(0) = F_0 \\
d\sigma_t &= \upsilon \sigma_t dW_t^{(2)}, \qquad \sigma(0) = \sigma_0 \\
dW_t^{(1)} dW_t^{(2)} &= \rho
\end{aligned} \tag{2}
$$

where F_t here is the $2Y5Y$ forward par swap rate.

b) Using initial values $\tilde{\sigma}_0 = 0.06$, $\tilde{\beta} = 0.5$, $\tilde{\upsilon} = 0.45$ and $\tilde{\rho} = -0.2$, fit the SABR model.

   i) Report the fitted parameter values $\hat{\sigma}_0$, $\hat{\beta}$, $\hat{\upsilon}$ and $\hat{\rho}$.

   ii) Assess whether the parameter values you have found match the observed market implied volatilities.

c) You are now approached by a client who wishes to protect himself against rising interest rates and for that purpose, he is interested in buying a digital option which pays 1 unit of currency in the event that the 5Y spot swap rate in exactly two years exceeds some strike K. The strike the client has in mind corresponds to $K_{offset} = 125$, meaning that the digital option will pay 1 if the 5Y par swap rate in 2 years is 125 bps or more above the 2Y5Y forward par swap rate you found in a). Recall that the 2Y5Y forward par swap rate will in exactly two years correspond to the spot 5Y par swap rate. To compute the fair value of this derivative, you decide to simulate σ_t and F_t in the SABR model using the fitted parameter values you have just found (If you were not able to fit the SABR model, you can use the suggested initial parameter values instead). The forward par swap rate, F, and the volatility σ can be simulated in the SABR model using a simple Euler scheme. Denote by M, the number of steps in the simulation and index the time points in the simulation by m, $m \in \{0, 1, 2, ..., M-1, M\}$ so that the time points will be $[t_0, t_1, ..., t_{M-1}, t_M] = [0, \delta, 2\delta, ..., T-\delta, T = \delta M]$ and hence the step in time will be of size $\delta = \frac{T}{M}$. The SABR model can then be simulated using the following equations

$$
\begin{aligned}
F_m &= F_{m-1} + \sigma_{m-1} F_{m-1}^{\beta} \sqrt{\delta} Z_m^{(1)}, \qquad F(0) = F_0 \\
\sigma_m &= \sigma_{m-1} + \upsilon \sigma_{m-1} \sqrt{\delta} \Big(\rho Z_m^{(1)} + \sqrt{1-\rho^2} Z_m^{(2)} \Big), \qquad \sigma(0) = \sigma_0
\end{aligned} \tag{3}
$$

where $Z_m^{(1)}$ and $Z_m^{(2)}$ are independent standard normal random variables. If we denote the *forward* par swap rate at present time $t = 0$ for an interest rate swap beginning at time T_1 and ending at time T_2 by $F_0(T_1, T_2)$, the digital option has at maturity $T_1 = 2$ a pay-off function $\chi(T_1)$ given by

$$
\chi(T_1) = \begin{cases} 1, & F_{T_1}(T_1, T_2) \geq F_0(T_1, T_2) + 0.0125 \\ 0, & F_{T_1}(T_1, T_2) < F_0(T_1, T_2) + 0.0125 \end{cases} \tag{4}
$$

To compute the time $t = 0$ price $\Pi(0)$ of the derivative, you can use that

$$
\Pi(0) = p(0, 2) \mathrm{E}^2 \big[\chi(T_1 = 2) \big] \tag{5}
$$

where E^2 is the two year forward measure under which the SABR model is defined and also simulated. To compute the fair price of this complex derivative, simulate the forward 5Y par swap rate over a two year period and repeat the simulation N times. Then compute the fraction of times the 5Y spot rate in exactly 2 years exceeds the strike and finally, once you have this fraction, you can multiply the fraction by $p(0, 2)$.

   i) Using at least $M = 2000$ time steps and at least $N = 10000$ simulations, but ideally a higher N, compute the fair value of the digital option.

ii) Discuss how well this digital option will protect the client against interest rate increases and in particular against drastic interest rate increases.

iii) Now, if you were to construct a derivative that resembles a digital option with the suggested strike and only had the swaptions listed above at your disposal, what swaptions would you include to construct the digital option and in what proportions?

Problem 3 - Solution

a) i) The par swap rate for the 2Y5Y forward swap is 0.04870.

ii) The plot can be found under b).

iii) There is indeed a 'smirk' in implied volatilities clearly indicating that market prices are not equivalent to what would arise in a Black's model. The pricing measure chosen by the market is *not* compatible with the $2Y5Y$ forward par swap rate following a log-normal distribution. The distribution implied by the measure chosen by the market has more fat tails and displays more left skewness than that of a log-normal random variable. This is a finding that is very much consistent with typical market behavior.

b) i) The fitted values will differ a bit depending on the method used to fit the parameters but they should be roughly $\hat{\sigma} \approx 0.05$, $\hat{\beta} \approx 0.7$, $\hat{\upsilon} \approx 0.65$ and $\hat{\rho} \approx -0.32$.

ii) The fit should be near perfect which can be demonstrated by reporting the mean-squared error between observed implied volatilities and fitted implied volatilities or by plotting observed and fitted implied volatilities.

c) i) The price of the digital option will of course be random as it is the result of simulation, but the price should be roughly 0.063

ii) The digital option will protect the client against rising interest rates but the pay-off of the digital option will be 1 no matter how far ITM the option is, so the client's pay-off won't be very high if interest rates rise sharply which, depending on his situation, could be a problem.

iii) The strategy that closest resembles a digital option with a strike 125 bps above the ATMF is to be long one maturity $K_{offset} = 100$ and short one $K_{offset} = 150$ swaption. The price of such an instrument would be roughly 14 bps.

Problem 4

A client of yours has just contacted you because he has a floating rate loan obligation on which he pays 6M EURIBOR semi-annully. The loan expires on todays date in exactly 6 years from now but the client is worried that 6M EURIBOR fixings after the next 3 years will be high. You therefore check market prices so that you can help your client. The 6M EURIBOR fixing has just been announced and your trusted colleague has just extracted the zero coupon spot rate curve and given you the numbers. In addition, you can also observe 6M EURIBOR interest rate caplet prices for a strike of 0.055 quoted in terms of Black implied volatility. The numbers are given in the table below.

Table 3: Continuously compounded spot rates and cap Black implied volatilities

T_i	0.50	1.00	1.50	2.00	2.50	3.00	3.50	4.00	4.50	5.00	5.50	6.00
$R(0, T_i)$	0.0371	0.0426	0.0469	0.0502	0.0528	0.0549	0.0564	0.0577	0.0586	0.0594	0.0600	0.0604
caplet $\bar{\sigma}_i$	-	0.182	0.205	0.229	0.255	0.283	0.313	0.347	0.381	0.423	0.484	0.545

a) Find zero coupon bond prices and 6M forward EURIBOR rates $L(0; T_{i-1}, T_i)$ that correspond to the continuously compounded spot rates.

i) Report, the 6M forward EURIBOR rates corresponding to the fixings taking place at T_{i-1} for $T_{i-1} = \{0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5\}$ in a table.

ii) Do the 6M forward EURIBOR rates already reflect that the market expects rising interest rates? How does that affect the cost to the client of protecting himself against rising interest rates?

You decide to offer your client two different options.

i) He can enter into a 3Y3Y forward payer swaption so that instead of paying a floating rate between year 3 and year 6 on his loan obligation, he will pay a fixed rate.

ii) He can enter into an interest rate cap with a strike of 0.055 that prevents his floating rate payments from exceeding 0.055 between year 3 and year 6.

b) i) Compute the 3Y3Y forward par swap rate the client would need to pay if he is to swap his floating rate payment from year 3 to year 6 into fixed rate payment.

ii) Using the caplets traded in the market, construct an interest cap that protects the client from paying more than 0.055 on his floating rate obligation. Compute the cost of such an arrangement both if the client pays for the cap upfront or if he pays in the form of a spread to be paid semi-annually over the next 6 years.

c) i) Discuss the upside and downside of each of these two arrangements and relate these to the cost of entering into each of these agreements.

ii) Do any of these agreements give him full protection against rising interest rates? If yes, why is that so? If no, how could he obtain full protection against rising interest rates.

Problem 4 - Solution

a) The 6M forward LIBOR rates can be computed directly to give us that

Table 4: 6M forward LIBOR rates

T_{i-1}	0.50	1.00	1.50	2.00	2.50	3.00	3.50	4.00	4.50	5.00	5.50
$L(0; T_{i-1}T_i)$	0.0374	0.0487	0.0563	0.0610	0.0642	0.0665	0.0679	0.0669	0.0677	0.0671	0.0659

ii) The market is already pricing for rising interest rates and that will needless to say be reflected in the cost of protecting against such rising interest rates.

b) i) The 3Y3Y forward par swap rate will be roughly 0.0670

ii) To construct the desired interest rate cap, the caplets with maturities ranging from $T = 3.5$ to $T = 6$ can be used. Individual Caplet prices can be found using Black's formula and the quoted caplet implied volatilities in the above. The upfront price of the interest rate cap becomes 0.0603 and the premium paid semi-annually for the next 6 years becomes roughly 60 bps.

c) i) Entering into the 3Y3Y forward interest eliminates all uncertainty about future interest rate payments but also comes with no upside should future interest rates be lower than feared. The other option of entering into an interest rate cap retains that upside but of course has the drawback that it, unlike the interest rate swap, comes at an upfront cost.

ii) Either of the two arrangements will protect the client from rising interest rates but only from year 3 and beyond. If he wants to be protected against rises in interest rates over the near term also, both of the two proposed solutions can be constructed to protect the client earlier at a higher cost in terms of lost upside potential in the case of the forward swap or higher upfront cost in terms of the interest rate cap.

Python Code

```
import numpy as np
from scipy.optimize import minimize
import fixed_income_derivatives as fid
import matplotlib.pyplot as plt

# Problem 1
r0, a, b, sigma = 0.034, 0.8, 0.042, 0.04
T_star = np.array([0.1,0.25,0.5,0.75,1,1.5,2,3,4,5,7,10])
R_star = fid.spot_rate_vasicek(r0, a, b, sigma, T_star)
R_star = np.round(R_star,4)

M = 13
sigma = 0.04
T_star = np.array([0.1,0.25,0.5,0.75,1,1.5,2,3,4,5,7,10])
R_star = np.array([0.0352, 0.0361, 0.0374, 0.0387, 0.0397, 0.0415, 0.0429, 0.0449, 0.0462, 0.0471, 0.0483, 0.0492])
# a)
param_0 = 0.038, 1.2, 0.035
result = minimize(fid.fit_vasicek_no_sigma_obj,param_0,method = 'nelder-mead',args = (sigma,R_star,T_star),options={'xatol': 1e-12,'disp': True})
r_0, a, b = result.x
print(f"Parameters from the fit. r_0: {r_0}, a: {a}, b: {b}, squared deviation: {result.fun}")
# b)
alpha = 0.5
T = np.array([alpha*i for i in range(0,13)])
p = fid.zcb_price_vasicek(r_0,a,b,sigma,T)
L = fid.zcb_to_forward_LIBOR_rates(T,p,horizon = 1)
S_swap = 0
for i in range(1,M):
    S_swap += alpha*p[i]
R_swap = (1-p[-1])/S_swap
print(f"par swap rate for a 6Y swap: {R_swap}")
# c)
price_caplet = np.zeros([M])
for i in range(2,M):
    price_caplet[i] = (1+alpha*R_swap)*fid.euro_option_price_vasicek(1/(1+alpha*R_swap),T[i-1],T[i],p[i-1],p[i],a,sigma,type = "put")
print(price_caplet)
price_cap = sum(price_caplet)
print(f"Price of the interest rate cap: {price_cap}, semi-annual premium = {alpha*price_cap/S_swap}")

# Problem 2
def idx_left_right_find(idx,indexes):
    idx_left, idx_right = None, None
    N = len(indexes)
    I_done = False
    n = 0
    while I_done is False and n < N - 3/2:
        if indexes[n] < idx and idx < indexes[n+1]:
            idx_left, idx_right = indexes[n],indexes[n+1]
            I_done = True
        n += 1
    return idx_left, idx_right

def spot_rate_inter_idx(idx_inter,idx_known,T,spot_rate,type = "linear"):
    if type == "linear":
        for idx in idx_inter:
            idx_left, idx_right = idx_left_right_find(idx,idx_known)
            if idx_left is not None and idx_right is not None:
                spot_rate[idx] = ((T[idx_right]-T[idx])*spot_rate[idx_left] + (T[idx]-T[idx_left])*spot_rate[idx_right])/(T[idx_right]-T[idx_left])
    return spot_rate

def spot_rate_inter_maturity(T,spot_rate_known,T_known,type_inter = "linear"):
    I_done = False
    if T < T_known[0]:
        if type_inter == "linear":
            spot_rate = ((T_known[1]-T)*spot_rate_known[0] + (T-T_known[0])*spot_rate_known[1])/(T_known[1]-T_known[0])
        I_done = True
    elif T_known[-1] < T:
        if type_inter == "linear":
            spot_rate = ((T_known[-1]-T)*spot_rate_known[-2] + (T-T_known[-2])*spot_rate_known[-1])/(T_known[-1]-T_known[-2])
        I_done = True
    else:
        i = 1
        while I_done is False and i < len(T_known) - 0.5:
            if T_known[i-1] <= T and T <= T_known[i]:
                spot_rate = ((T_known[i]-T)*spot_rate_known[i-1] + (T-T_known[i-1])*spot_rate_known[i])/(T_known[i]-T_known[i-1])
            i += 1
    return spot_rate

def rates_insert(rate_insert,indexes,rate):
    j = 0
    for idx in indexes:
        rate[idx] = rate_insert[j]
        j += 1
    return rate

def accrual_factor(idx_maturity,p,idx_swap_fixed):
    accrual_factor = 0
    j, I_done = 0,False
    while I_done == False and j < len(idx_swap_fixed):
        if idx_swap_fixed[j] < idx_maturity + 0.5:
            accrual_factor += p[idx_swap_fixed[j]]
            j += 1
        else:
            I_done = True
    accrual_factor *= delta_swap_fixed
    return accrual_factor

def zcb_curve_cali_obj(spot_rate_knot,idx_swap_knot,spot_rate,idx_inter,idx_knot,idx_swap_fixed,swap_rate_market,scaling = 10000,type_inter = "linear"):
    spot_rate = rates_insert(spot_rate_knot,idx_swap_knot,spot_rate)  # inserting candidate spotrates into vector of all spot rates
    spot_rate = spot_rate_inter_idx(idx_inter,idx_knot,T,spot_rate,type = type_inter)
    p = np.zeros([len(spot_rate)])
    for idx in idx_swap_fixed:
```

```python
        p[idx] = np.exp(-spot_rate[idx]*T[idx])
    swap_rate = np.zeros([len(spot_rate_knot)])
    i = 0
    for idx in idx_swap_knot:
        S = accrual_factor(idx,p,idx_swap_fixed)
        swap_rate[i] = (1-p[idx])/S
        i += 1
    mse = 0
    for i in range(0,len(swap_rate_market)):
        mse += (swap_rate_market[i] - swap_rate[i])**2
    mse *= scaling/len(swap_rate_market)
    return mse

def zcb_yield_curve_cali(spot_rate,L,fra_rate_market,swap_rate_market,T,indices,scaling = 10000,type_inter = "linear"):
    idx_libor, idx_fra_knot, idx_fra_inter, idx_swap_fixed, idx_swap_knot, idx_swap_inter, idx_inter, idx_knot = indices
    spot_rate[idx_libor] = np.log(1+L*(T[idx_libor]-T[0]))/(T[idx_libor]-T[0])
    spot_rate[2*idx_libor] = (spot_rate[idx_libor]*(T[idx_libor]-T[0]) + fra_rate_market[idx_libor-1]*(T[2*idx_libor]-T[idx_libor]))/(T[2*idx_libor]-T[0])
    args = (idx_swap_knot,spot_rate,idx_swap_inter,idx_knot,idx_swap_fixed,swap_rate_market,scaling,type_inter)
    result = minimize(zcb_curve_cali_obj,swap_rate_market,args = args,options={'disp': False})
    print(f"Spot rates at knot points after minimization: {result.x}")
    # Inserting computed spot rates and interpolating to all time points where there is a cashflow for some swap
    for i in range(0,len(idx_swap_knot)):
        spot_rate[idx_swap_knot[i]] = result.x[i]
    spot_rate = spot_rate_inter_idx(idx_inter,idx_knot,T,spot_rate,type = type_inter)
    # Computing spot rates for maturities before the LIBOR fixing implied by the FRA's
    for i in range(1,idx_libor):
        spot_rate[i] = (spot_rate[i+idx_libor]*T[i+idx_libor] - np.log(1+fra_rate_market[i-1]*T[idx_libor]))/T[i]
    return spot_rate

alpha = 1/2
type_inter = "linear"
scaling = 10000
L_6M = 0.04110
idx_libor = 6

M_fra = 16
fra_rate_market = [0.04358, 0.04423, 0.04484, 0.04542, 0.04597, 0.04649, 0.04698, 0.04743, 0.04787]
delta_fra = 1/12

M_swap = 58
delta_swap_fixed = 1/2
swap_rate_market = [0.04723, 0.04891, 0.05004, 0.05084, 0.05184, 0.05263, 0.05325, 0.05354, 0.05382]

T = np.array([i*delta_fra for i in range(0,M_fra)] + [3/2 + delta_swap_fixed*i for i in range(0,M_swap)])
spot_rate_known = np.zeros([M_fra+M_swap])
spot_rate_known[idx_libor] = np.log(1+L_6M*(T[idx_libor]-T[0]))/(T[idx_libor]-T[0])
spot_rate_known[2*idx_libor] = (spot_rate_known[idx_libor]*(T[idx_libor]-T[0]) + fra_rate_market[idx_libor-1]*(T[2*idx_libor]-T[idx_libor]))/(T[2*idx_libor]-T[0])

idx_all = set([i for i in range(0,M_fra+M_swap)])
idx_known = set([idx_libor,2*idx_libor])
idx_fra_knot = idx_known.union(set({M_fra+1}))
idx_fra_inter = set([i for i in range(idx_libor,M_fra)]).symmetric_difference(idx_fra_knot)
idx_swap_fixed = set({idx_libor,2*idx_libor}).union(set([i for i in range(M_fra,M_fra+M_swap)]))
idx_swap_knot = set({17,19,21,23,27,33,43,53,73})
idx_swap_inter = idx_swap_fixed.symmetric_difference(idx_swap_knot)
idx_knot = idx_fra_knot.union(idx_swap_knot)
idx_inter = idx_fra_inter.union(idx_swap_inter)

idx_fra_knot = sorted(idx_fra_knot,reverse = False)
idx_fra_inter = sorted(idx_fra_inter,reverse = False)
idx_swap_fixed = sorted(idx_swap_fixed,reverse = False)
idx_swap_knot = sorted(idx_swap_knot,reverse = False)
idx_swap_inter = sorted(idx_swap_inter,reverse = False)
idx_inter = sorted(idx_inter,reverse = False)
idx_knot = sorted(idx_knot,reverse = False)

# Finding spot rates at knot points based on swap market data
args = (idx_swap_knot,spot_rate_known.copy(),idx_swap_inter,idx_knot,idx_swap_fixed,swap_rate_market,scaling,type_inter)
result = minimize(zcb_curve_cali_obj,swap_rate_market,args = args,options={'disp': False})
# print(f"Spot rates at knot points after minimization:")
# print(result.x)

# Inserting computed spot rates and interpolating to all time points where there is a cashflow for some swap
spot_rate = spot_rate_known.copy()
for i in range(0,len(idx_swap_knot)):
    spot_rate[idx_swap_knot[i]] = result.x[i]
spot_rate = spot_rate_inter_idx(idx_inter,idx_knot,T,spot_rate,type = type_inter)
# Computing spot rates for maturities before the LIBOR fixing implied by the FRA's
for i in range(1,idx_libor):
    spot_rate[i] = (spot_rate[i+idx_libor]*T[i+idx_libor] - np.log(1+fra_rate_market[i-1]*T[idx_libor]))/T[i]
p = fid.spot_rates_to_zcb(T,spot_rate)
print(f"spot_rates. 0.5Y: {spot_rate[6]}, 1Y: {spot_rate[12]}, 3Y: {spot_rate[19]}, 5Y: {spot_rate[23]}, 10Y: {spot_rate[33]}, 20Y: {spot_rate[53]}, 30Y: {spot_rate[73]}")

# b)
T_swap, p_swap = np.zeros([60]), np.zeros([60])
i = 0
for idx in idx_swap_fixed:
    T_swap[i] = T[idx]
    p_swap[i] = p[idx]
    i += 1
R_swap, S_swap = np.zeros([60]), np.zeros([60])
for i in range(1,60):
    S_swap[i] = alpha*sum(p_swap[:i+1])
    R_swap[i] = (1-p_swap[i])/S_swap[i]

# c)
idx_6Y_swap = 11
K = 0.047
print(f"6Y par swap rate: {R_swap[idx_6Y_swap]}. S_swap: {S_swap[idx_6Y_swap]}, Value of 6Y receiver swap: {(K-R_swap[idx_6Y_swap])*S_swap[idx_6Y_swap]}")

fig = plt.figure(constrained_layout=False, dpi = 300, figsize = (5,3))
fig.suptitle(f"Calibrated zero coupon spot rates", fontsize = 9)
gs = fig.add_gridspec(nrows=1,ncols=1,left=0.12,bottom=0.2,right=0.88,top=0.90,wspace=0,hspace=0)
```

```
ax = fig.add_subplot(gs[0,0])
xticks = [0,1,2,3,4,5,7,10,15,20,30]
ax.set_xticks(xticks)
ax.set_xticklabels(xticks,fontsize = 6)
ax.set_xlim([xticks[0]+-0.2,xticks[-1]+0.2])
plt.xlabel(f"Maturity",fontsize = 6)
ax.set_yticks([0,0.02,0.04,0.06])
ax.set_yticklabels([0,0.02,0.04,0.06],fontsize = 6)
ax.set_ylim([0,0.061])
plt.grid(axis = 'y', which='major', color=(0.7,0.7,0.7,0), linestyle='--')
p1 = ax.scatter(T[1:], spot_rate[1:], s = 1, color = 'black', marker = ".",label="Spot rates")
p2 = ax.scatter(T_swap[1:], R_swap[1:], s = 1, color = 'red', marker = ".",label="Par swap rates")
plots = [p1,p2]
labels = [item.get_label() for item in plots]
ax.legend(plots,labels,loc="lower right",fontsize = 6)

# Problem 3
alpha = 0.5
T_max = 7
idx_exer, idx_set = 4, 14
K_swaption_offset = [-300,-250,-200,-150,-100,-50,0,50,100,150,200,250,300]
N_swaption = len(K_swaption_offset)
M = int(round(T_max/alpha)+1)
T = np.array([i*alpha for i in range(0,M)])
p = np.array([1, 0.98314916, 0.96478677, 0.94539738, 0.92535353, 0.90493951, 0.88437071, 0.86380916, 0.84337571, 0.8231596, 0.80322594, 0.78362142, 0.76437872, 0.74551992, 0.72705911])
price_market = np.array([0.12256859, 0.10253932, 0.08273803, 0.0633625, 0.04480655, 0.02793572, 0.0145331, 0.00650867, 0.0030062, 0.00158778, 0.00094974, 0.00062285, 0.00043427])

S_swap = 0
for i in range(idx_exer+1,idx_set + 1):
    S_swap += alpha*p[i]
R_swap = (p[idx_exer] - p[idx_set])/S_swap
print(f"accrual factor: {S_swap}, 2Y5Y par swap rate: {R_swap}")
K, iv_market = np.zeros([N_swaption]), np.zeros([N_swaption])
for i in range(0,N_swaption):
    K[i] = R_swap + K_swaption_offset[i]/10000
    iv_market[i] = fid.black_swaption_iv(price_market[i],T[idx_exer],K[i],S_swap,R_swap,type = "call", iv0 = 0.25, max_iter = 1000, prec = 1.0e-12)
print(iv_market)

param_0 = 0.06, 0.5, 0.45,-0.2
result = minimize(fid.fit_sabr_obj,param_0,method = 'nelder-mead',args = (iv_market,K,T[idx_exer],R_swap),options={'xatol': 1e-8,'disp': True})
print(f"Parameters from fit: {result.x}, squared dev: {result.fun}")
sigma_0_fit, beta_fit, upsilon_fit, rho_fit = result.x
iv_fit, price_fit = np.zeros([N_swaption]), np.zeros([N_swaption])
for i in range(0,N_swaption):
    iv_fit[i] = fid.sigma_sabr(K[i],T[idx_exer],R_swap,sigma_0_fit,beta_fit,upsilon_fit,rho_fit,type = "call")
    price_fit[i] = fid.black_swaption_price(iv_fit[i],T[idx_exer],K[i],S_swap,R_swap,type = "call")
print(f"Implied volatilities from market prices:")
print(iv_fit)

# Plot of market implied volatilities
fig = plt.figure(constrained_layout=False, dpi = 300, figsize = (5,3))
fig.suptitle(f"Market implied volatilities", fontsize = 10)
gs = fig.add_gridspec(nrows=1,ncols=1,left=0.12,bottom=0.2,right=0.88,top=0.90,wspace=0,hspace=0)
ax = fig.add_subplot(gs[0,0])
xticks = K_swaption_offset
ax.set_xticks(xticks)
ax.set_xticklabels(xticks,fontsize = 6)
ax.set_xlim([xticks[0]-10,xticks[-1]+10])
plt.xlabel(f"Strike offset",fontsize = 7)
ax.set_yticks([0,0.1,0.2,0.3,0.4])
ax.set_yticklabels([0,0.1,0.2,0.3,0.4],fontsize = 6)
ax.set_ylim([0,0.408])
ax.set_ylabel(f"Implied volatility",fontsize = 7)
plt.grid(axis = 'y', which='major', color=(0.7,0.7,0.7,0), linestyle='--')
p1 = ax.scatter(K_swaption_offset, iv_market, s = 6, color = 'black', marker = ".",label="IV market")
p2 = ax.scatter(K_swaption_offset, iv_fit, s = 1, color = 'red', marker = ".",label="IV fit")
plots = [p1,p2]
labels = [item.get_label() for item in plots]
ax.legend(plots,labels,loc="upper right",fontsize = 5)
# fig.savefig("C:/Jacob/Uni_of_CPH/Interest rate derivatives/final_exam_2023_version_B/problem_3a.pdf")
# plt.show()

M_simul, T_digital = 2000, 2
N_simul = 100001
strike = R_swap + 125/10000
F_simul, sigma_simul = np.zeros([M_simul]), np.zeros([M_simul])
frac_ITM = np.zeros([N_simul])
count = 0
for n in range(0,N_simul):
    F_simul, sigma_simul = fid.sabr_simul(R_swap, sigma_0_fit, beta_fit, upsilon_fit, rho_fit, M_simul, T_digital)
    if F_simul[-1] > strike:
        count += 1
    frac_ITM[n] = count/(n+1)
print(f"frac ITM: {frac_ITM[-1]}, price = {p[4]*frac_ITM[-1]}")
idx_conv_check = np.array([1000*i for i in range(0,101)])
for idx in idx_conv_check:
    print(frac_ITM[idx])

# Problem 4
M = 13
alpha = 0.5
T = np.array([i*alpha for i in range(0,M)])
R = 0.055
spot_rate = np.array([np.nan,0.0371, 0.0426, 0.0469, 0.0502, 0.0528, 0.0549, 0.0564, 0.0577, 0.0586, 0.0594, 0.0600, 0.0604])
sigma_caplet = np.array([np.nan,np.nan,0.182,0.205,0.229,0.255,0.283,0.313,0.347,0.381,0.423,0.484,0.545])

# a)
p = fid.spot_rates_to_zcb(T,spot_rate)
L = fid.zcb_to_forward_LIBOR_rates(T,p,horizon = 1)
print(f"6M froward EURIBOR rates")
print(L)

# b)
```

```
S_swap = 0
for i in range(7,M):
    S_swap += alpha*p[i]
R_swap = (p[6]-p[-1])/S_swap
print(f"par swap rate for a 3Y3Y swap: {R_swap}")
price_caplet = np.zeros([M])
for i in range(2,M):
    price_caplet[i] = fid.black_caplet_price(sigma_caplet[i],T[i],R,alpha,p[i],L[i],type = "call")
print(f"price_caplets: {price_caplet}")
price_cap = sum(price_caplet[7:])
print(f"price_cap: {price_cap}")
S_cap = 0
for i in range(1,M):
    S_cap += alpha*p[i]
R_cap = price_cap/S_cap
print(f"R_cap: {R_cap}, premium twice a year: {alpha*R_cap}")
```