Fixed Income Derivatives Final Exam Fall 2024(22.02.2025)

Throughout this exam you will rely on a term structure of continuously compounded zero coupon bond spot rates and thus equivalently, a term structure of zero coupon bond prices. These term structures will be the result of a term structure fit that you will be asked to perform in Problem 1. The fit will be based on data for the 6M Euribor fixing just announced, Forward Rate Agreements(FRA) and interest rate swap(IRS) par swap rates. These swaps will all involve a floating leg paying 6 months Euribor and a fixed leg paying annual coupons. The data for the 6M Euribor fixing, the FRA's and the par swap rates can be found below in Table 1.

Table 1: Euribor, FRA and Swap Market Data

Euribor	Fixing	FRA	Midquote	IRS	Midquote
6M	0.02869	1X7	0.03075	2Y	0.04329
		2X8	0.03273	3Y	0.04936
		3X9	0.03463	4Y	0.05349
		4X10	0.03645	5Y	0.05622
		5X11	0.03820	7Y	0.05898
		6X12	0.03988	10Y	0.05966
		7X13	0.04148	15Y	0.05797
		8X14	0.04302	20Y	0.05599
		9X15	0.04449	30Y	0.05334

At the end of this document, a Python script is presented containing the data from the table above. You do not have to use this piece of code, but it might save you some time.

Problem 1

In this problem, we will fit a term structure of continuously compounded zero coupon bond spot rates to the market data we have from Table 1 above. Using the method you believe is most suitable, fit the term structure of continuously compounded ZCB spot rates.

a) Discuss which characteristics the fitted term structures of spot- and forward rates should have.

b) Plot the term structures of fitted spot- and forward rates from your fit for maturities up to 30 years. Explain which method you used and determine if the term structures from your fit have the desired characteristics described in Problem 1a.

c) Compute and report the par swap rates of the 2Y, 5Y, 8Y, 10Y and 30Y interest rate swaps paying 6M floating Euribor against a fixed rate paid annually based on your fitted spot rates. How can you check if the par swap rates you have computed are correct, and what do these values reveal about the quality of your fitted ZCB spot rates.

d) Compute 6M forward Euribor rates and include these in the plot from above. The 6M forward Euribor rates and the continuously compounded forward rates should be close but not quite the same. Briefly discuss why that is the case.

Problem 1 - Solution

a) The characteristics that need to be imposed on the term structures of ZCB prices and forward rates depends a bit on the pricing problem at hand but as a minimum, spot rates should be continuous and forward rates should be positive. In many cases, it is however necessary to impose further conditions such as forward rates being differentiable.

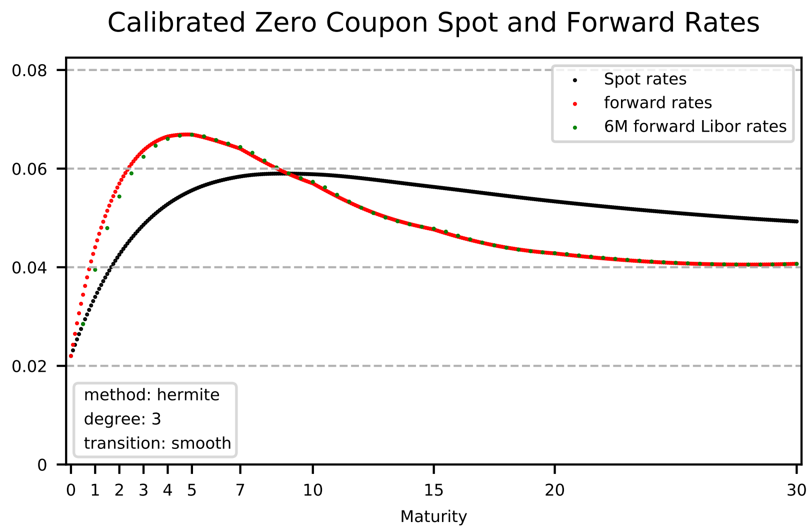

Figure 1: Fitted Term Structure of ZCB Spot Rates

b) To fit the term structure of ZCB spot rates, third degree polynomials with a 'smooth' transition were used and the fitted term structures can be seen in Figure 1. Using a transition of 'smooth' guarantees that forward rates are differentiable so they are for this fit as well.

c) The par swap rates for the maturities 2Y, 5Y, 8Y, 10Y and 30Y become

Table 2: Par Swap Rates Based on the ZCB Fit

T_i	2	5	8	10	30
R_{swap}	0.04329	0.05622	0.05952	0.05966	0.05334

To check that the par swap rates we have computed are correct, remember that the algorithm we used to fit our ZCB term structure is designed so that the resulting term structure fit is consistent with Euribor, FRA and par swap rates observed in the market. So, if this fit is successful, the computed par swap rates should be identical to the par swap rates observed in the market. It should thus be that the 2Y, 5Y, 10Y and 30Y par swap rates from Table 2 are identical to the corresponding market rates, and we see that they are, confirming that our ZCB term structure fit is indeed consistent with market data.

d) The term structure of 6M forward Euribor rates are also included in Figure 1. The fact that forward rates and 6M forward rates differ stems from the fact that the forward rate is in essence a continuously compounded rate over an infinitely small time interval and the 6M forward Euribor rate is simple rate compounded over a discrete time interval. This difference can also be seen from their respective definitions given below.

$$\begin{aligned} f(t,T) &= -\frac{\partial \log p(t,T)}{\partial T}, \\ L(t;T_{i-1},T_i) &= -\frac{p(t,T_i) - p(t,T_{i-1})}{(T_i - T_{i-1})p(t,T_i)}. \end{aligned} \tag{1}$$

Problem 2

In this problem, we consider three fixed income instruments: A 6Y receiver swap on which the holder pays 6M Euribor and receives an annual fixed rate, a 6Y fixed rate bond paying an annual coupon of 0.06 and a 6Y zero coupon bond. Assume that all three instruments have a principal of $K = 100$ Euros. For these three assets, we will investigate their sensitivity towards changes in the term structure of zero coupon bond spot rates.

a) First, we will compute some key figures for all instruments.

   i) For the 6Y receiver swap, find and report its par swap rate and the accrual factor.

   ii) For the 6Y fixed rate bond, find and report its price, yield-to-maturity, duration and convexity.

   iii) For the 6Y zero coupon bond, find its price, yield-to-maturity, duration and convexity.

b) Then, we will compute the exact gain or loss to the three instruments if *all* ZCB spot rates were to rise by 10 basispoints.

   i) Compute the exact absolute gain or loss to the 6Y receiver swap if *all* ZCB spot rates were to rise by 10 basispoints.

   ii) Compute the exact absolute gain or loss to the 6Y fixed rate bond if *all* ZCB spot rates were to rise by 10 basispoints.

   iii) Compute the exact absolute gain or loss to the 6Y zero coupon bond if *all* ZCB spot rates were to rise by 10 basispoints.

c) Now, we will investigate the sensitivity of all three instruments to individual points on the ZCB spot rate curve.

   i) Compute one-by-one the absolute gain or loss to the 6Y receiver swap if each of the 1Y, 2Y, 3Y, 4Y, 5Y and 6Y zero coupon spot rates were to rise by 10 basispoints and report the gain or loss in each of these 6 scenarios.

   ii) Compute one-by-one the absolute gain or loss to the 6Y fixed rate bond if each of the 1Y, 2Y, 3Y, 4Y, 5Y and 6Y zero coupon spot rates were to rise by 10 basispoints and report the gain or loss in each of these 6 scenarios.

   iii) Compute one-by-one the absolute gain or loss to the 6Y zero coupon bond if each of the 1Y, 2Y, 3Y, 4Y, 5Y and 6Y zero coupon spot rates were to rise by 10 basispoints and report the gain or loss in each of these 6 scenarios.

d) Finally, we will discuss the exposure you get from the 6Y receiver swap, the 6Y fixed rate bond and the 6Y zero coupon bond respectively.

   i) Briefly explain which point on the ZCB spot rate curve the 6Y fixed rate bond and the 6Y zero coupon bond are most exposed to, and please explain why that is so.

   ii) The 6Y receiver swap does not involve the exchange of principal at maturity. Yet, you are likely to have found that the 6Y receiver swap gives you much exposure to the 6Y spot rate. Briefly explain why that is so.

   iii) Please compare the risk exposure of the three instruments and determine which, if any, of three instruments that are quite similar and which, if any, that are very different.

Problem 2 - Solution

a) The key figures for the three instruments become.

   i) 6Y receiver swap: $R_{swap} = 0.05794$ and $S_0^6 = 5.02511$.

   ii) 6Y Fixed rate bond: $\Pi_{fr} = 101.0359$, $y_{fr} = 0.05791$, $D_{fr} = 5.21697$ $K_{fr} = 29.54187$.

   iii) 6Y Zero Coupon Bond: $\Pi_{zcb} = 70.8853$, $y_{zcb} = 0.05903$, $D_{zcb} = 6$ $K_{zcb} = 36$.

b) The absolute gains to each of the three instruments if ZCB spot rates were to rise by 10 basispoints. become.

   i) 6Y receiver swap: -0.52045.

   ii) 6Y Fixed rate bond: -0.52388.

   iii) 6Y Zero Coupon Bond: -0.42404.

c) The sensitivities of the three instruments to the 1Y, 2Y, 3Y, 4Y, 5Y and 6Y ZCB spot rates become

Table 3: Sensitivities of the Three Instruments to ZCB Spot Rates

T_i	1	2	3	4	5	6
i) Swap	-0.0056	-0.01063	-0.01500	-0.01873	-0.02188	-0.44861
ii) FRB	-0.0058	-0.01101	-0.01553	-0.01939	-0.02266	-0.44948
iii) ZCB	0	0	0	0	0	-0.42404

d) The table of sensitivities in Table 3 reveals the nature of the exposure of the three fixed income instruments.

   i) Almost all of the fixed rate bonds exposure and the zero coupon bonds exposure is towards changes in the 6Y ZCB spot rate. In general a fixed income instrument will be most exposed to the ZCB spot rates for maturities that correspond to the largest discounted future cashflow of the fixed income instrument. The fixed rate bond has a large cashflow at maturity in 6 years when the principal is repaid and likewise for the zero coupon bond which only has one cashflow at maturity.

   ii) Now, the 6Y receiver swap does *not* involve an exchange of principal at maturity and yet, almost all of the 6Y receiver swap's exposure comes from the 6Y spot rate. The explanation for this seemingly surprising conclusion can be found in the floating leg. Using the usual replication argument, it can be shown that the value V_{fl} of the floating leg is a function of the 6Y spot rate denoted $R(0,6)$ and that V_{fl} is given by

$$V_{fl} = K_{fl}\big(1 - p(0,6)\big) = K_{fl}\Big(1 - e^{6 \cdot R(0,6)}\Big) \tag{2}$$

   iii) Altogether, the three instruments are fairly similar in the sense that they give the investor exposure to the 6Y spot rate. Of course there are some differences and in particular that the ZCB bond gives you only exposure to the 6Y spot rate, whereas the other two give you some exposure to spot rates with shorter maturities.

Problem 3

Imagine that you are working for a financial institution and one of your clients has approached you to help him manage the risk of 10Y floating rate obligation on which the client must pay 6M Euribor beginning in exactly 6 months. That is, every 6 months, at time T_i say, the client pays an interest rate of $L(T_{i-1}, T_i)$ on his obligation, where $L(T_{i-1}, T_i)$ is the Euribor rate announced 6 months prior. For simplicity, we will set the principal of this obligation to $K = 1$ such that your answers will be per unit of the debt obligation principal. The client wishes to protect himself against the Euribor rate rising above $R_{cap} = 0.07$, so he has asked for a quote on a 10Y interest rate cap consisting of a series of caplets each with a payoff $\chi_{caplet}(T_i)$ at time T_i given by

$$\chi_{caplet}(T_i) = \big(L(T_{i-1}, T_i) - R_{cap}\big)_+ \tag{3}$$

per unit of debt principal, where $L(T_{i-1}, T_i)$ is the 6M Euribor rate announced 6 months earlier at time T_{i-1}. Now, in order to reduce the cost to the client, you have proposed that he could also take a short position in an interest rate floor which would imply that the future interest rate payments of the client cannot go below some strike R_{floor}. Recall that much like an interest rate cap, an interest rate floor consists of a series of floorlets each with a payoff $\chi_{floorlet}(T_i)$ a time T_i given by

$$\chi_{floorlet}(T_i) = \big(R_{floor} - L(T_{i-1}, T_i)\big)_+ \tag{4}$$

per unit of debt principal. A caplet can be seen as a European call option with $L(T_{i-1}, T_i)$ as the underlying asset and similarly, a floorlet can be seen as a put option with $L(T_{i-1}, T_i)$ as the underlying asset.
In order to find a fair value of the interest rate cap and the interest rate floor, we will use two different models and compare the outcomes. First, we will use the Vasicek model in which the dynamics of the short rate r_t are given by

$$dr_t = (b - ar_t)dt + \sigma dW_t, \qquad r(0) = r_0 \tag{5}$$

where $a > 0$, $b > 0$ and $\sigma \in \mathbb{R}$ are all constant parameters, W_t is a Brownian motion and r_0 is the known initial value of the short rate. Second, we will use the Ho-Lee model in which the dynamics of the short rate are

$$dr_t = \Theta(t)dt + \sigma dW_t, \qquad r(0) = r_0 \tag{6}$$

where $\sigma \in \mathbb{R}$ is a constant parameter, $\Theta(t)$ is a deterministic function, W_t is a Brownian motion and r_0 is the known initial value of the short rate. Also assume for both models that $\sigma = 0.01$.

a) We will begin by fitting both the Vasicek model and the Ho-Lee model to the term structure of fitted zero coupon bond spot rates you found in Problem 1.

   i) Using the initial parameter values $r_0 = 0.025$, $a = 3$, $b = 0.1$, and of course also using that we know $\sigma = 0.01$, fit a Vasicek model to the term structure of fitted zero coupon bond spot rates you found in Problem 1. Plot the term structure of ZCB spot rates in your fitted version of the Vasicek model together with the ZCB spot rates from Problem 1. Comment on the quality of the Vasicek model fit and in particular, assess if the Vasicek model is a good choice of model given the shape of the ZCB term structure you found in Problem 1.

   ii) Fit the Ho-Lee model to the term structures you found in Problem 1 by using that we know $\sigma = 0.01$ and then choosing the function $\Theta(t)$ so that it is consistent with forward rates. Also, explain how well the Ho-Lee model fits this initial ZCB spot rate term structure and why that is so. Finally, plot $\Theta(t)$ and relate the shape of $\Theta(t)$ to the shape of the term structure of ZCB spot rates from Problem 1.

b) Next, we will simulate the short rates in both the Vasicek and Ho-Lee models using the parameters we obtained in Problem 3a. Your simulation should run from time $t = 0$ to $t = 10$ years and should involve taking at least $M = 1000$ steps. Also, please choose the same 'seed' both when simulating the Vasicek and the Ho-Lee model. If you were not able to perform the fit in the previous problem, please proceed using the Vasicek model and the parameter values suggested as starting values.

i) Using the parameter values of the fitted Vaicek model, simulate one trajectory of the short rate. Plot the simulated trajectory as well as the mean of the short rate and a two-sided 95 percent confidence interval. Comment on the size of the confidence interval as a function of time and explain why the confidence interval behaves the way it does. Also, assess if the dynamics of the short rate in the Vasicek model seem realistic.

ii) Using the function $\Theta(t)$, simulate a trajectory of the short rate in the fitted Ho-Lee model. Also include a two-sided confidence interval in your plot, comment on the size of the confidence interval as a function of time and explain why the confidence interval behaves the way it does. Also, assess if the dynamics of the short rate in the Ho-Lee model seem realistic.

c) Now, we will compute the price of an interest rate cap with a strike of $R_{cap} = 0.07$ and the price of an interest rate floor with a strike of $R_{floor} = 0.05$. We will do so both using the Vasicek model as well as the Ho-Lee model and compare the results.

i) Compute caplet prices using both the fitted Vasicek model as well as the fitted Ho-Lee model and report the caplet prices in basispoints for maturities $T_i \in [2, 4, 6, 8, 10]$. Then compute the price of the 10Y interest rate cap with a strike of $R_{cap} = 0.07$ for both models and report the price of the cap in basispoints if paid upfront.

ii) The price of a floorlet can, just as a caplet, be reexpressed as a European option with $p(T_{i-1}, T_i)$ as the underlying asset. Which type of European option on $p(T_{i-1}, T_i)$ is the floorlet equivalent to and what is the strike of this option?

iii) Write functions in Python that allow you to compute floorlet prices in both the Vasicek and Ho-Lee models and use these functions to compute floorlet prices with a strike of $R_{floor} = 0.05$ in both models. Report floorlet prices in basispoints for $T_i \in [2, 4, 6, 8, 10]$ and also report the prices of the interest rate floors for both models.

iv) Finally, try to find the strike R^*_{floor} in both the Vasicek and Ho-Lee models that would allow the client to reduce the total cost of managing his interest rate exposure to 0 provided that he enters into a long position in the interest rate cap with a strike of $R_{cap} = 0.07$ and a short position into an interest rate floor with a strike of R^*_{floor}. To solve this problem, you should set up an optimization problem, but you can also use trial-and-error if that will save you time.

d) Lastly, we will compare the prices of interest rate caps and floors in our two models.

i) Discuss and compare the fitted Vasicek model and the fitted Ho-Lee model from Problem 3b. Explain the strengths and weaknesses of these two models. Also suggest a model that would mitigate some of the shortcomings these two models might have while retaining some of the positive features.

ii) You are likely to have found that there is substantial difference in the prices of the cap and floor between the two models. Explain the role σ plays in the pricing of interest rate caps and floors, and explain why these two models lead to very different cap and floor prices, despite σ being the same in both models.

iii) In Problem 3c, you were able to construct a way for the client to both achieve his goal of putting an upper bound on future interest rate payments and reduce the cost of doing so to 0. Explain why that is so, and in particular relate your answer to the potential upside and downside risk the client faces taking into account his total position including the debt obligation, the long position in the $R_{cap} = 0.07$ interest rate cap and the short position in the R^*_{floor} interest rate floor.

Problem 3 - Solution

a) Fitting both the Vasicek and the Ho-Lee models and plotting the resulting fitted term structures results in the plots shown in Figure 2 below.

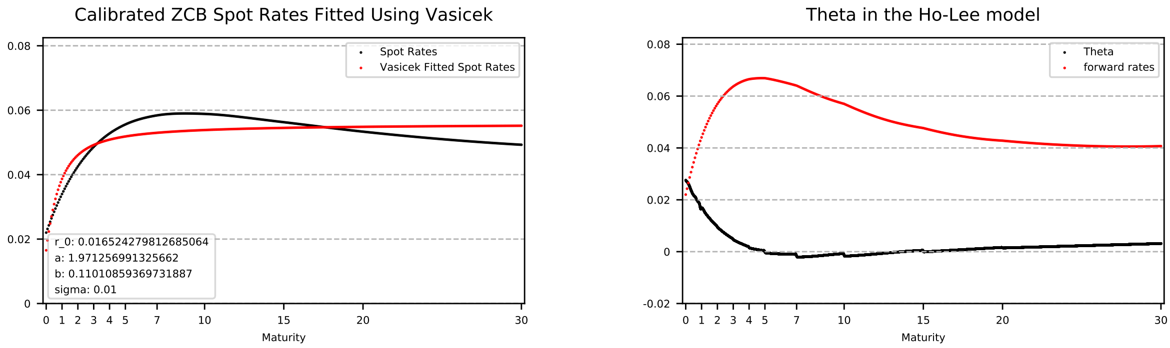

Figure 2: Vasicek Model Fit and Theta in the Ho-Lee Model

i) The fitted parameters in the Vasicek model become roughly $\hat{r}_0$.016524, $\hat{a} = 1.971257$ and $\hat{b} = 0.110109$. From the plot in Figure 2, it is quite apparent that the Vasicek model is not able to fit the term structure of ZCB spot rates we found in Problem 1. This is however no surprise as the ZCB spot rate term structure has a 'hump' and a Vasicek model can not fit such a term structure. In fact, it can be shown that spot rates in the Vasicek model are either strictly increasing or strictly decreasing and that property is clearly not compatible with the market spot rates we are observing. A more flexible model able to fit an initial term structure that has a hump is therefore needed, and such a model could for example be the Ho-Lee model.

ii) The Ho-Lee model can by construction fit any initial term structure of spot rates, provided forward rates are differentiable, and can therefore also fit the term structure we have here. The reason why the Ho-Lee model has this flexibility is that the function $\Theta(t)$ takes a different value for each t and is thus infinitely dimensional. The way to insure that the Ho-Lee model fits the initial term structure is to set

$$\Theta(t) = \frac{\partial f^*(0,t)}{\partial T} + \sigma^2 t = f^*_T(0,t) + \sigma^2 t. \tag{7}$$

Here, $f^*(0,t)$ is the observed forward rate for maturity t and the derivative $\frac{\partial f^*(0,t)}{\partial T}$ in (7) is to be understood as the derivative with respect to the second argument of f^* evaluated at maturity t. The function $\Theta(t)$ in the Ho-Lee model controls the drift of the short rate as a function of time and therefore has an interpretation in terms of the shape of the ZCB forward rate term structure. For maturities at which forward rates are, rising $\Theta(t)$ is positive which is also apparent when comparing Figures 1 and 2, where we see that $\Theta(t) > 0$ up to maturities of roughly 4-5 years corresponding to rising forward rates. For maturities immediately beyond 5 years $\Theta(t) < 0$ while forward rates decline but $\Theta(t)$ turns positive again for maturities above 15 years as the term $\sigma^2 t$ in (7) becomes more influential.

b) The trajectories of the short rate in the Vasicek and the Ho-Lee models are simulated along with their means and two-sided confidence intervals, and the result is shown in Figure 3 below.

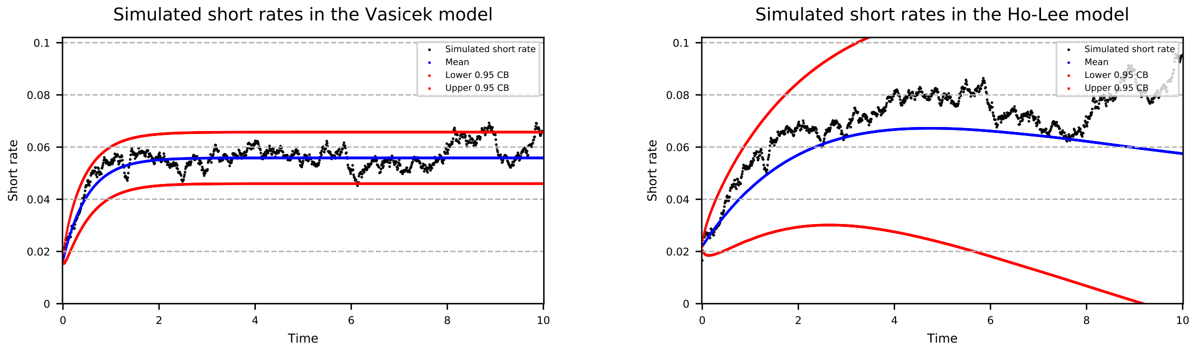

Figure 3: Simulated short rates in the Vasicek and Ho-Lee models

i) Looking at the simulated trajectory of the short rate in the Vasicek model and its confidence interval, we notice that the confidence interval settles very quickly, after roughly two years, reflecting that the short rate is mean-reverting in the Vasicek model and that the distribution of the short rate has settled to the stationary distribution. However, it is quite clear that the confidence interval is unrealistically narrow. The size of the confidence interval under the stationary distribution is a constant times $\frac{\sigma}{\sqrt{a}}$ but in an attempt to fit the hump-shaped term structure of spot rates, a is forced to take a relatively large value compared to σ^2 resulting in a confidence interval that becomes very narrow.

ii) From the plot of the simulated trajectory of the short rate and its confidence interval in the Ho-Lee model, we notice that the confidence intervals become very large and are ever expanding in size. This is due to the fact that the short rate in the Ho-Lee model is not mean-reverting, and the distribution of the short rate does not settle to a stationary distribution. The size of the confidence interval in the Ho-Lee model is increasing over time and evolves as $\sigma\sqrt{t}$. The trajectory of the short rate can therefore become quite 'wild' after some time, and though the Ho-Lee model by construction fits the initial term structure, the short rate in this model does not behave very realistically over the long term.

c) Caplet prices in both models are computed and the price of the 10Y interest rate cap with a strike of $R_{cap} = 0.07$ can be found as the sum of caplet prices.

i) Caplet prices for maturities $T_i = 2, 4, 6, 8, 10$ in the two models are shown in Table 4 below.

Table 4: Caplet Prices

T_i	1	2	4	6	8	10
Vasicek(bps.)	0	0.00001	1.43273	1.09421	0.04516	0.00038
Ho-Lee(bps.)	0.00013	3.38223	25.78555	29.05696	24.86976	21.18938

The price of the $R_{cap} = 0.07$ interest rate cap in the Vasicek model becomes 10.39450 basispoints and in the Ho-Lee model the price of the cap becomes 389.83910 basispoints.

ii) The payoff of a floorlet can be expressed in terms of zero coupon bond prices by using the usual replication argument that allows us to replicate future Euribor fixings. The replication argument implies that

$$L(T_{i-1}, T_i) = \frac{1 - P(T_{i-1}, T_i)}{\delta P(T_{i-1}, T_i)} \tag{8}$$

where $\delta = T_i - T_{i-1}$. The payoff of the floorlet at time T_i can then be written as

$$\begin{aligned}\chi_{floorlet}(T_i) =& \big(R_{floor} - L(T_{i-1}, T_i)\big)_+ = \delta\left[R_{floor} - \frac{1 - P(T_{i-1}, T_i)}{\delta P(T_{i-1}, T_i)}\right]_+ \\ =& \frac{1 + \delta R_{floor}}{p(T_{i-1}, T_i)}\left[p(T_{i-1}, T_i) - \frac{1}{1 + \delta R_{floor}}\right]_+ \end{aligned} \tag{9}$$

Discounting back to time T_{i-1} by multiplying by $p(T_{i,T_i})$ gives us that the payoff from the floorlet is equivalent to a time T_{i-1} payoff of

$$\big(1 + \delta R_{floor}\big)\left[p(T_{i-1}, T_i) - \frac{1}{1 + \delta R_{floor}}\right]_+ \tag{10}$$

We recognize this payoff as that of the payoff to $\big(1 + \delta R_{floor}\big)$ European call options with a strike of $\frac{1}{1+\delta R_{floor}}$ and $p(T_{i-1}, T_i)$ as the underlying asset.

iii) The Python function that can be used to compute prices of floorlets in the Vasicek and Ho-Lee models can be found in Appendix A. The floorlet prices for a strike of $R_{floor} = 0.05$ become

Table 5: Floorlet Prices

T_i	1	2	4	6	8	10
Vasicek(bps.)	48.91082	0.39521	0	0	0.00016	0.01967
Ho-Lee(bps.)	50.19115	13.21356	7.97818	12.23255	18.85744	24.90057

The price of the $R_{floor} = 0.05$ interest rate floor in the Vasicek model becomes 59.59678 basispoints and the price of the $R_{floor} = 0.05$ interest rate floor in the Ho-Lee model becomes 317.13915 basispoints.

iv) To find the strike R^*_{floor} of the floor that would allow the client to manage his interest rate risk at no cost when having a long position in the $R_{cap} = 0.07$ interest rate cap and a short position in the R^*_{floor}, we effectively need to find the R^*_{floor} for which the price of the interest rate cap and the price of the floor are the same. This can be done by setting up a minimization problem where the objective function is a function of R^*_{floor} that returns the squared difference between Π_{cap} and $\Pi_{floor}\big(R^*_{floor}\big)$. The code for the objective function is given in Appendix A. We get that $R^*_{floor} = 0.04142$ in the Vasicek model and $R^*_{floor} = 0.05286$ in the Ho-Lee model.

d) Clearly, prices of caps and floors differ greatly between the Vaicek and Ho-Lee models. We will now explain why that is.

i) The Vasicek model was not able to fit the hump in the ZCB spot rates and in attempt to do so forced the parameter a governing mean reversion to assume a relatively large value. This resulted in a Vasicek model fit in which the short rate settles very quickly to a distribution with very narrow confidence bands. This in tuen resulted in very low prices of both the interest rate cap and the interest rate floor. The Ho-Lee model, on the other hand, will by construction fit the initial term structure but has no stationary distribution and the confidence bands are increasing over time. This resulted in very high prices of the interest rate cap and the interest rate floor. Now, prices of interest rate caps and floors are strictly speaking not a function of $\Theta(t)$, but for consistency, it is best to work with a model that fits the initial term structure of interest rates. It is however also an advantage to work with a model that has a stationary distribution to get more realistic prices of derivatives with maturities far in the future. An alternative model that could solve the problems faced by both the Vasicek and the Ho-Lee model is the Hull-White Extended Vasicek model. In this model, you get both a perfect fit of the initial term structure as well as a stationary distribution.

ii) Yes, we did indeed find that cap and floor prices were much higher in the Vasicek model. Now, prices of caps and floors depend on prices of caplets and floorlets respectively which are essentially European options with future ZCB prices as their underlying assets. European call- and put prices depend positively on σ so prices of caps and floors do as well. However, the fact that the short rate is mean-reverting with narrow confidence bands in the Vasicek model and that the short rate does not settle to a stationary distribution in the Ho-Lee model, confidence bands are in fact ever expanding, implies that European option prices on zero coupon bonds grow much faster as a function of maturity in the Ho-Lee model.

iii) It might seem counterintuitive that the client is in fact able to mange his interest risk and prevent that he will ever pay more than $R_{cap} = 0.07$ on his floating rate obligation by entering into a long position in a $R_{cap} = 0.07$ and a short position in a R^*_{floor} interest rate floor. However, taking a short position implies that you receive a payment upfront and to extent that the value of the long and short position combined is zero initially, there is no time zero cashflow and the entire strategy comes at zero cost. The client does get protection from downside risk by capping his interest rate payments. However, he also foregoes some upside potential by entering into the short position in the interest rate floor. This short position implies that the client has to pay the buyer of the interest rate floor when 6M Euribor rates are below the strike meaning that the interest rate payments of the client cannot go below the strike of the floor regardless of low interest rate become.

Problem 4

Next, we will work towards computing the price of a certain type of 'lookback' option where the underlying is the 5 year forward 5 year par swap rate denoted $F(t) = R_n^N(t)$ where $T_n = 5$ and $T_N = 10$. In order to price this option, we will assume that the underlying forward par swap rate F_t follows a SABR model. Hence, we assume that F_t has stochastic volatility σ_t and both quantities are driven by Brownian motion. F_t and σ_t have the following joint dynamics

$$
\begin{aligned}
dF_t &= \sigma_t F_t^{\beta} dW_t^{(1)}, \qquad && F(0) = F_0, \\
d\sigma_t &= \upsilon \sigma_t dW_t^{(2)}, && \sigma(0) = \sigma_0, \\
dW_t^{(1)} dW_t^{(2)} &= \rho dt
\end{aligned}
\tag{11}
$$

where $0 \leq \beta \leq 1$, $0 < \upsilon$ and $-1 < \rho < 1$ are constants, and $W_t^{(1)}$ and $W_t^{(2)}$ are correlated Brownian motions with correlation coefficient ρ.
The lookback option we will consider has a strike K_{lb} and gives the owner a payoff at maturity T_n that equals the difference between the strike K_{lb} and smallest value that the underlying forward par swap rate has taken between present time $t = 0$ and maturity at $t = T_n$. The payoff of the lookback option can in other words be written as

$$
\chi(T_n) = K_{lb} - \min_{0 \leq t \leq T_n} F_t \tag{12}
$$

To find a suitable model for the 5 year forward 5 year par swap rate, we will use data consisting of market implied volatilities for 5Y5Y payer swaptions given in Table 6 below.

Table 6: $5Y5Y$ Swaption Market Implied Volatilities

$K_{offset}(bp)$	-300	-250	-200	-150	-100	-50	ATMF
$\sigma_{swaption}$	0.275616	0.237815	0.209661	0.183214	0.163977	0.140387	0.132907
$K_{offset}(bp)$	50	100	150	200	250	300	-
$\sigma_{swaption}$	0.124768	0.128710	0.134726	0.145571	0.149179	0.160561	-

a) First, we will plot the market implied volatility as a function of strike denoted K.

i) Find and report the 5Y5Y forward par swap rate denoted R_5^5 based on your ZCB term structure fit from Problem 1 and hence also find the strike 'ATMF'(At-the-money-forward) in Table 6 above that corresponds to the 5Y5Y forward par swap rate.

ii) Plot market implied volatilities as a function of the strike K. Is the shape of implied volatility as a function of K in line with what is common in financial markets? What can be said about the shape of the distribution of the underlying par swap rate implied by the measure chosen by market participants and expressed in the shape of implied volatility as a function of K?

b) In this question, we will fit the SABR model to the market implied volatilities from Table 6. In doing so, you can assume we know from other sources that $\beta = 0.58$ and use the initial parameter values $\tilde{\sigma}_0 = 0.032$, $\tilde{\upsilon} = 0.6$ and $\tilde{\rho} = -0.20$ to calibrate the SABR model to the market implied volatilities.

i) Report the parameter values you obtain when fitting the model as well as the SSE of your fit.

ii) Plot the fitted market implied volatilities in the same plot as the observed market implied volatilities and assess if you were able to fit market data.

To compute the price of the lookback option, we will need to simulate the underlying forward par swap rate. Recall that the forward rate F_t and the volatility σ_t can be simulated using an Euler scheme. Denote by M, the number of steps in the simulation and index the time points in the simulation by m,

$m \in \{0, 1, 2, ..., M-1, M\}$. The time points will then be $[t_0, t_1, ..., t_{M-1}, t_M = T_n] = [0, \delta, 2\delta, ..., T_n - \delta, T_n = \delta M]$ and hence the step in time will be of size $\delta = \frac{T_n}{M}$. The model can then be simulated using the following equations

$$\begin{aligned} F_m &= F_{m-1} + \sigma_{m-1} F_{m-1}^{\beta} \sqrt{\delta} Z_m^{(1)}, && F(0) = F_0 \\ \sigma_m &= \sigma_{m-1} + \upsilon \sigma_{m-1} \sqrt{\delta} \Big(\rho Z_m^{(1)} + \sqrt{1-\rho^2} Z_m^{(2)} \Big) = \sigma_{m-1} + \upsilon \sigma_{m-1} \sqrt{\delta} Z_m^{(3)}, && \sigma(0) = \sigma_0 \end{aligned} \tag{13}$$

where $Z_m^{(1)}$ and $Z_m^{(2)}$ are independent standard normal random variables. Hence, $Z_m^{(3)}$ is also a standard normal random variable with $\mathrm{Cor}\big[Z_m^{(1)}, Z_m^{(3)}\big] = \rho$ and $\mathrm{Cor}\big[Z_m^{(2)}, Z_m^{(3)}\big] = \sqrt{1-\rho^2}$. Also, we have that $Z_m^{(i)}$ and $Z_u^{(j)}$ are independent for $m \neq u$ and all $i, j \in [1, 2, 3]$.

c) Now, we will simulate F_t and σ_t in the SABR model from $t = 0$ to $t = T_n = 5$ by taking $M = 5000$ steps, using the parameter values you found in your fit and the 5Y5Y forward par swap rate you found in Problem 4ai as the initial value of the forward rate. If for some reason you were unable to find reasonable parameter values, you can use the suggested starting values from Problem 4b instead.

   i) As we know, the Euler scheme can be improved and a Milstein scheme performs better numerically. Write difference equations for F_m and σ_m according to a Milstein scheme and also write a function in Python that allows you to simulate F_t and σ_t using a Milstein scheme.

   ii) It is well known that the trajectories of F_t and σ_t are not always very realistic in the SABR model. Perform 4 simulations for different choices of 'seed' and plot the trajectories of both F_t and σ_t, preferably in the same plot with F_t on the left axis and σ_t on the right axis. In one of the plots, the trajectories of F_t and σ_t should seem realistic and in the other three, try to show examples of unrealistic trajectories of F_t and σ_t. For each of these three plots, explain why the trajectories are not realistic.

d) Finally, we will price the lookback option for a strike $K_{lb} = R_5^5$ equal to the 5Y5Y forward par swap rate you found in Problem 4ai. To do so, we will need to simulate the trajectory of the underlying 5Y5Y forward par swap rate H times where H should be at least 2000 and M should be 5000. Also, we will need to discount the payoff in each simulation back to time $t = 0$. If you were not able to construct a Milstein scheme in Problem 3c, you can use the Euler scheme suggested in 13.

   i) Explain why discounting can be done using one of the observed ZCB prices and present an expression for the price of the lookback option involving this discount factor. Report the numerical value of the discount factor.

   ii) Find and report an estimate of the price of the lookback option in basispoints and plot the estimated price as a function of the number of trajectories performed. For what H can you reasonably say that the price estimate has converged?

   iii) In which scenarios will the lookback option result in a large payoff? Discuss which type of exposure the owner of the lookback option will get and which type of investor might be interested in such an option.

Problem 4 - Solution

a) We begin by fitting the SABR model to observed market implied volatilities.

  i) The 5Y5Y forward rate is $R_5^5 = 0.06439$ and hence, the strike of the ATMF strike of the 5Y5Y swaption is $K_{ATMF} = 0.06439$.

  ii) Swaption implied volatilities have the shape of an asymmetric 'smile' or perhaps 'smirk' which is very common for observed option implied volatilities. The higher implied volatilities for deep in-the-money and deep out-of-the-money swaptions reflect that the implied distribution of the underlying par swap rate under the pricing measure chosen by the market generally has more probability mass in the tales of the distribution than that of the log-normal distribution. The asymmetry and much higher implied volatilities for small strikes reflects that the distribution under the pricing measure is not symmetric but left-skewed and has even more probability mass in the left tail of the distribution.

b) The SABR is now fitted to market implied volatilities for the 5Y5Y swaption.

  i) The parameter estimates become roughly $\hat{\sigma}_0 = 0.03581$, $\hat{\upsilon} = 0.65415$ and $\hat{\rho} = -0.25319$.

  ii) A plot of the market and fitted implied volatilities is shown in Figure 4. We see that the fitted values are close to observed values, and we conclude that the SABR model with these parameters (as well as $\beta = 0.58$) fits market implied volatilities quite well.

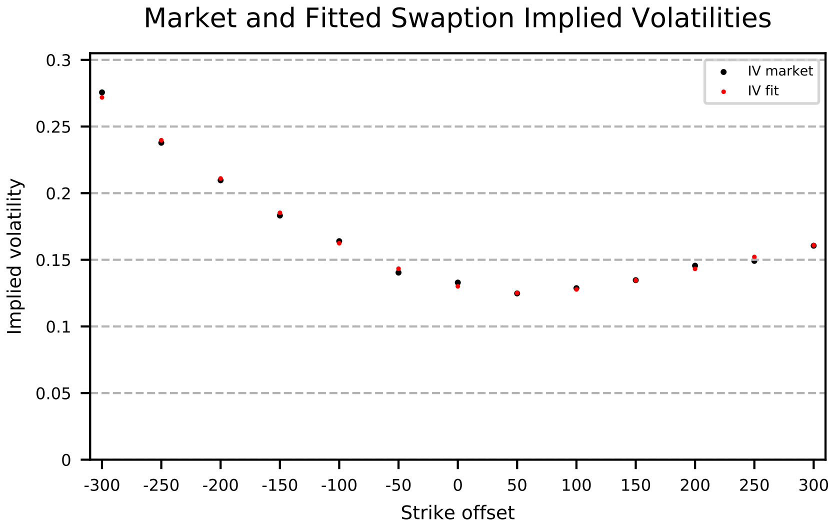

Figure 4: Market and Fitted Implied Volatility of the 5Y5Y Swaption

c) Next, we simulate the forward rate and the volatility in the SABR model using the fitted parameter values from Problem 4b.

  i) First, we will need to create a Milstein scheme to simulate the forward rate F and the volatility σ. The difference equations for F_m and σ_m using a Milstein scheme are given below.

$$
\begin{aligned}
F_m &= F_{m-1} + \sigma_{m-1} F_{m-1}^{\beta} \sqrt{\delta} Z_m^{(1)} + \frac{1}{2} \beta \sigma_{m-1}^2 F_{m-1}^{2\beta-1} \delta \Big(Z_m^{(1)^2} - 1 \Big), && F(0) = F_0, \\
\sigma_m &= \sigma_{m-1} + \upsilon \sigma_{m-1} \sqrt{\delta} Z_m^{(3)} + \frac{1}{2} \upsilon^2 \sigma_{m-1} \delta \Big(Z_m^{(3)^2} - 1 \Big), && \sigma(0) = \sigma_0. \qquad (14)
\end{aligned}
$$

  ii) Four different trajectories of F_t and σ_t are shown in Figure 5 below. In the top left plot, the forward rate and its volatility behave rather 'normally'. In the top right plot, volatility becomes very small and the forward rate freezes. In the bottom left plot, volatility increases drastically and eventually, the forward rate blows up. In the bottom right plot, volatility also blows up, but this time, the forward rate goes to zero.

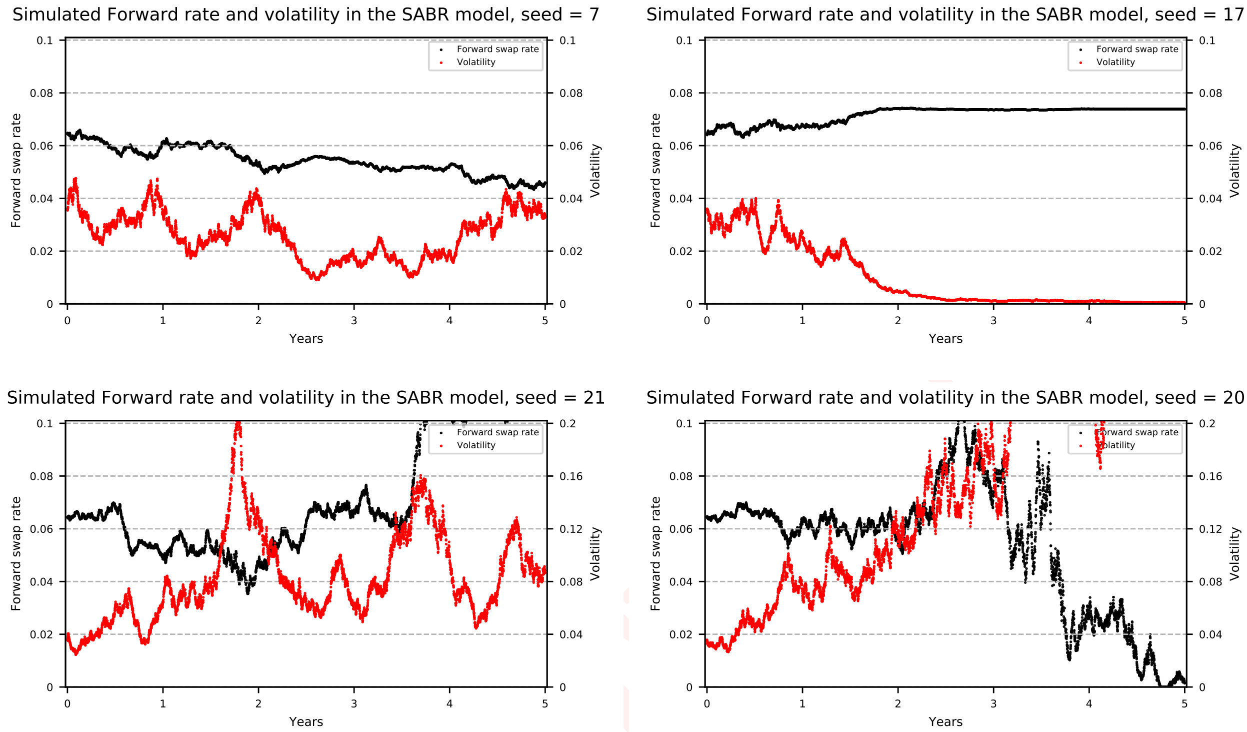

Figure 5: Simulations of the Forward Rate and Volatility in the SABR model

d) We now turn to estimating the price of the lookback option.

i) The 5Y5Y forward rate serving as the underlying of the 5Y5Y payer swaption is a martingale under the measure $\mathbb{Q}^{T_n}$ where $p(t, T_n)$ is the numeraire. Denoting the price of the lookback option by $\Pi(t)$, we have that $\frac{\Pi(t)}{p(t,T_n)}$ is a martingale under $\mathbb{Q}^{T_n}$. We can therefore write that

$$\frac{\Pi(0)}{p(0,T_n)} = \mathbb{E}^{T_n}\left[\frac{\Pi(T_n)}{p(T_n,T_n)}\right] = \mathbb{E}^{T_n}\left[\Pi(T_n)\right]. \tag{15}$$

Now, the price of the lookback option at maturity $T_n = 5$ is just the payoff of the lookback option and we have that

$$\Pi(0) = p(0,T_n)\mathbb{E}^{T_n}\left[K_{lb} - \min_{0\leq t\leq T_n} F_t\right] \tag{16}$$

The price of the lookback option can therefore be estimated by simulating H trajectories of the underlying forward rate to estimate the expectation in (16) and then discounting back to time $t = 0$ using the zero coupon price $p(0, 5) = 0.75734$.

ii) A plot of the price of the lookback option as a function of the number of simulations is shown in Figure 6. The price seems the have settled after roughly 4000 simulations reaching a final price after 5000 simulations of roughly $\Pi(0) = 99.466$ basispoints.

iii) The lookback option will finish far in the money if the future forward rate drops very far below the initial value of the forward rate. This is more likely to happen if the volatility of the forward rate is high. The lookback option is therefore attractive to investor who would like to either hedge against falling interest rates or speculate that interest rates will fall. Furthermore, the lookback options also gives the investor exposure to interest rate volatility.

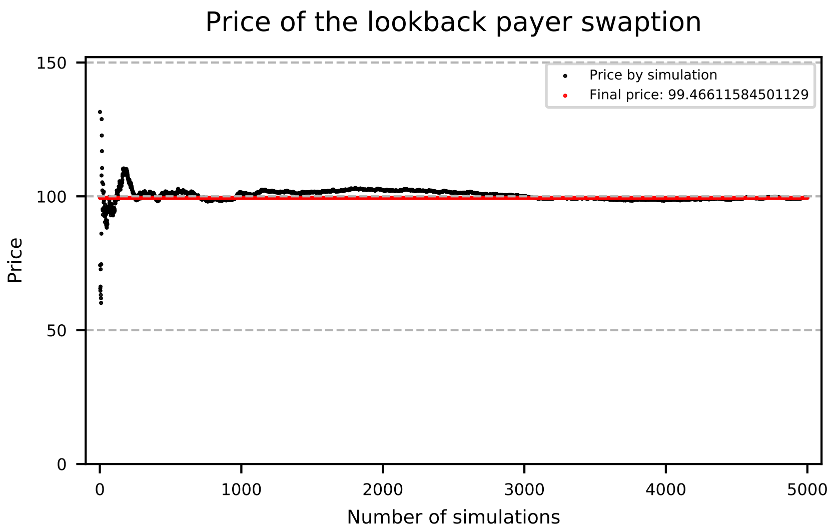

Figure 6: Price of the Lookback Option

A Python Code

```
import numpy as np
from scipy.optimize import minimize
import fixed_income_derivatives_E2024 as fid
import matplotlib.pyplot as plt

iv_swaption_market = np.array([0.275616, 0.237815, 0.209661, 0.183214, 0.163977, 0.140387, 0.132907, 0.124768, 0.12871, 0.134726, 0.145571, 0.149179, 0.160561])
K_swaption_offset = np.array([-300,-250,-200,-150,-100,-50,0,50,100,150,200,250,300])

Euribor_fixing = [{"id": 0,"instrument": "libor","maturity": 1/2, "rate": 0.02869}]
fra_market = [{"id": 1,"instrument": "fra","exercise": 1/12,"maturity": 7/12, "rate": 0.03075},
{"id": 2,"instrument": "fra","exercise": 2/12,"maturity": 8/12, "rate": 0.03273},
{"id": 3,"instrument": "fra","exercise": 3/12,"maturity": 9/12, "rate": 0.03463},
{"id": 4,"instrument": "fra","exercise": 4/12,"maturity": 10/12, "rate": 0.03645},
{"id": 5,"instrument": "fra","exercise": 5/12,"maturity": 11/12, "rate": 0.03820},
{"id": 6,"instrument": "fra","exercise": 6/12,"maturity": 12/12, "rate": 0.03988},
{"id": 7,"instrument": "fra","exercise": 7/12,"maturity": 13/12, "rate": 0.04148},
{"id": 8,"instrument": "fra","exercise": 8/12,"maturity": 14/12, "rate": 0.04302},
{"id": 9,"instrument": "fra","exercise": 9/12,"maturity": 15/12, "rate": 0.04449}]
swap_market = [{"id": 10,"instrument": "swap","maturity": 2, "rate": 0.04329, "float_freq": "semiannual", "fixed_freq": "annual","indices": []},
{"id": 11,"instrument": "swap","maturity": 3, "rate": 0.04936, "float_freq": "semiannual", "fixed_freq": "annual","indices": []},
{"id": 12,"instrument": "swap","maturity": 4, "rate": 0.05349, "float_freq": "semiannual", "fixed_freq": "annual","indices": []},
{"id": 13,"instrument": "swap","maturity": 5, "rate": 0.05622, "float_freq": "semiannual", "fixed_freq": "annual","indices": []},
{"id": 14,"instrument": "swap","maturity": 7, "rate": 0.05898, "float_freq": "semiannual", "fixed_freq": "annual","indices": []},
{"id": 15,"instrument": "swap","maturity": 10, "rate": 0.05966, "float_freq": "semiannual", "fixed_freq": "annual","indices": []},
{"id": 16,"instrument": "swap","maturity": 15, "rate": 0.05797, "float_freq": "semiannual", "fixed_freq": "annual","indices": []},
{"id": 17,"instrument": "swap","maturity": 20, "rate": 0.05599, "float_freq": "semiannual", "fixed_freq": "annual","indices": []},
{"id": 18,"instrument": "swap","maturity": 30, "rate": 0.05334, "float_freq": "semiannual", "fixed_freq": "annual","indices": []}]
data_zcb = Euribor_fixing + fra_market + swap_market

# Problem 1a - Fitting a ZCB term structure to the market data.
mesh = 1/12
M = 360
T_fit, R_fit = fid.zcb_curve_fit(data_zcb,interpolation_options = interpolation_options)
T_inter = np.array([i*mesh for i in range(0,M+1)])
p_inter, R_inter, f_inter, T_inter = fid.zcb_curve_interpolate(T_inter,T_fit,R_fit,interpolation_options = interpolation_options)
R_2Y_swap, S_2Y_swap = fid.swap_rate_from_zcb_prices(0,0,2,"annual",T_inter,p_inter)
R_5Y_swap, S_5Y_swap = fid.swap_rate_from_zcb_prices(0,0,5,"annual",T_inter,p_inter)
R_8Y_swap, S_80Y_swap = fid.swap_rate_from_zcb_prices(0,0,8,"annual",T_inter,p_inter)
R_10Y_swap, S_10Y_swap = fid.swap_rate_from_zcb_prices(0,0,10,"annual",T_inter,p_inter)
R_30Y_swap, S_30Y_swap = fid.swap_rate_from_zcb_prices(0,0,30,"annual",T_inter,p_inter)
print(f"1b - R_2Y_swap: {R_2Y_swap}, R_5Y_swap: {R_5Y_swap}, R_8Y_swap: {R_8Y_swap}, R_10Y_swap: {R_10Y_swap}, R_30Y_swap: {R_30Y_swap}")
T_6M = np.array([i*0.5 for i in range(0,61)])
p_6M = np.array(fid.for_values_in_list_find_value_return_value(T_6M,T_inter,p_inter))
L_6M = fid.forward_rates_from_zcb_prices(T_6M,p_6M,horizon = 1)

# Problem 2
K_6Y_swap, mat_6Y_swap = 100, 6
K_frb, mat_frb, cp_rate_frb = 100, 6, 0.06
K_zcb, mat_zcb = 100, 6
delta_zcb_rates = 0.001
R_6Y_swap, S_6Y_swap = fid.swap_rate_from_zcb_prices(0,0,mat_6Y_swap,"annual",T_inter,p_inter)
print(f"2ai - 6Y swap. R_6Y_swap: {R_6Y_swap}, S_6Y_swap: {S_6Y_swap}")
T_frb = np.array([i for i in range(0,mat_frb+1)])
C_frb = np.zeros(mat_frb + 1)
for i in range(1,mat_frb + 1):
    C_frb[i] = (T_frb[i]-T_frb[i-1])*cp_rate_frb*K_frb
C_frb[-1] += K_frb
p_frb = np.array(fid.for_values_in_list_find_value_return_value(T_frb,T_inter,p_inter))
price_frb = np.matmul(p_frb,C_frb)
ytm_frb = fid.ytm(price_frb,T_frb,C_frb)
D_frb = fid.macauley_duration(price_frb,T_frb,C_frb,ytm_frb)
conv_frb = fid.convexity(price_frb,T_frb,C_frb,ytm_frb)
print(f"2aii - 6Y Fixed rate bond. Price: {price_frb}, YTM: {ytm_frb}, D_frb: {D_frb}, K_frb: {conv_frb}")
p_zcb = fid.for_values_in_list_find_value_return_value(mat_zcb,T_inter,p_inter)
price_zcb = p_zcb*K_zcb
ytm_zcb = (1/p_zcb)**(1/mat_zcb)-1
D_zcb = mat_zcb
conv_zcb = mat_zcb**2
print(f"2aiii - 6Y ZCB. Price: {price_zcb}, YTM: {ytm_zcb}, D_zcb: {D_zcb}, K_zcb: {conv_zcb}")
# Problem 2b
R_inter_shift = R_inter + np.array([delta_zcb_rates for i in range(0,len(R_inter))])
p_inter_shift = fid.zcb_prices_from_spot_rates(R_inter_shift,T_inter)
R_6Y_swap_shift, S_6Y_swap_shift = fid.swap_rate_from_zcb_prices(0,0,mat_6Y_swap,"annual",T_inter,p_inter_shift)
change_6Y_swap_exact = (R_6Y_swap-R_6Y_swap_shift)*S_6Y_swap_shift*K_6Y_swap
print(f"2bi - 6Y swap after shift. R: {R_6Y_swap_shift}, S: {S_6Y_swap_shift}. Exact absolute change using the accrual factor: {change_6Y_swap_exact}")
p_frb_shift = np.array(fid.for_values_in_list_find_value_return_value(T_frb,T_inter,p_inter_shift))
price_frb_shift = np.matmul(p_frb_shift,C_frb)
ytm_frb_shift = fid.ytm(price_frb_shift,T_frb,C_frb)
print(f"2bii - 6Y Fixed rate bond after shift. YTM: {ytm_frb_shift}. Exact absolute change: {price_frb_shift-price_frb}")
p_zcb_shift = fid.for_values_in_list_find_value_return_value(mat_zcb,T_inter,p_inter_shift)
price_zcb_shift = p_zcb_shift*K_zcb
ytm_zcb_shift = (1/p_zcb_shift)**(1/mat_zcb)-1
print(f"2biii - 6Y ZCB after shift. YTM: {ytm_zcb_shift}. Exact absolute change: {price_zcb_shift-price_zcb}")
# Problem 2c
idx_shift = np.array([12,24,36,48,60,72])
change_6Y_swap, change_frb, change_zcb = np.zeros(6), np.zeros(6), np.zeros(6)
for i, idx in enumerate(idx_shift):
    R_inter_shift = R_inter.copy()
    R_inter_shift[idx] = R_inter[idx] + delta_zcb_rates
    p_inter_shift = fid.zcb_prices_from_spot_rates(R_inter_shift,T_inter)
    R_6Y_swap_shift, S_6Y_swap_shift = fid.swap_rate_from_zcb_prices(0,0,mat_6Y_swap,"annual",T_inter,p_inter_shift)
    change_6Y_swap[i] = (R_6Y_swap-R_6Y_swap_shift)*S_6Y_swap_shift*K_6Y_swap
    p_frb_shift = np.array(fid.for_values_in_list_find_value_return_value(T_frb,T_inter,p_inter_shift))
    price_frb_shift = np.matmul(p_frb_shift,C_frb)
    change_frb[i] = price_frb_shift-price_frb
    p_zcb_shift = fid.for_values_in_list_find_value_return_value(mat_zcb,T_inter,p_inter_shift)
    price_zcb_shift = p_zcb_shift*K_zcb
    change_zcb[i] = price_zcb_shift-price_zcb
print(f"2ci - 6Y swap. Sensitivities: {np.round(change_6Y_swap,5)}, {sum(change_6Y_swap)}")
print(f"2cii - Fixed rate bond. Sensitivities: {np.round(change_frb,5)}, {sum(change_frb)}")
print(f"2ciii - Zero coupon bond. Sensitivities: {np.round(change_zcb,5)}, {sum(change_zcb)}")
```

```
# Problem 3
strike_cap = 0.07
strike_floor = 0.05
mat_caplet = 10
param_0 = 0.025, 3, 0.1
sigma_vasicek = 0.01
sigma_ho_lee = 0.01
result = minimize(fid.fit_vasicek_no_sigma_obj,param_0,method = 'nelder-mead',args = (sigma_vasicek,R_inter,T_inter),options={'xatol': 1e-20,'disp': False})
r0_vasicek, a_vasicek, b_vasicek = result.x
print(f"3ai - Vaseicek parameters: r0: {r0_vasicek}, a: {a_vasicek}, b: {b_vasicek}, sigma: {sigma_vasicek}, SSE: {result.fun}")
R_vasicek = fid.spot_rate_vasicek(r0_vasicek,a_vasicek,b_vasicek,sigma_vasicek,T_inter)
f_fit = np.array(fid.for_values_in_list_find_value_return_value(T_fit,T_inter,f_inter))
mesh_theta, M_theta = 1/1000, 30000
T_star = np.array([i*mesh_theta for i in range(0,M_theta+1)])
f_star, f_T_star = fid.interpolate(T_star,T_inter,f_inter,interpolation_options)
theta = fid.theta_ho_lee(T_star,sigma_ho_lee,method = "default",f_T=f_T_star)
# Problem 3b - Simulation
M_simul, T_simul, seed = 1000, 10, 5
mesh_simul = T_simul/M_simul
size_ci = 0.95
r0 = r0_vasicek
t_simul = np.array([i*mesh_simul for i in range(0,M_simul+1)])
mean_vasicek = fid.mean_vasicek(r0,a_vasicek,b_vasicek,sigma_vasicek,t_simul)
lb_vasicek, ub_vasicek = fid.ci_vasicek(r0,a_vasicek,b_vasicek,sigma_vasicek,t_simul,size_ci,type_ci = "two_sided")
np.random.seed(seed)
r_vasicek = fid.simul_vasicek(r0,a_vasicek,b_vasicek,sigma_vasicek,M_simul,T_simul,method = "exact",seed = seed)
f_simul, f_T_simul = fid.interpolate(t_simul,T_fit,f_fit,interpolation_options)
np.random.seed(seed)
r_ho_lee = fid.simul_ho_lee(r0,f_T_simul,sigma_ho_lee,T_simul,method = "exact",f=f_simul,seed=seed)
mean_ho_lee, var_ho_lee = fid.mean_var_ho_lee(f_simul,sigma_ho_lee,t_simul)
lb_ho_lee, ub_ho_lee = fid.ci_ho_lee(f_simul,sigma_ho_lee,t_simul,size_ci,type_ci = "two_sided")
# Problem 3c - Cap and floor prices
T_caplet = np.array([i*0.5 for i in range(0,int(mat_caplet/0.5)+1)])
p_caplet = np.array(fid.for_values_in_list_find_value_return_value(T_caplet,T_inter,p_inter))
price_caplet_vasicek = fid.caplet_prices_vasicek(sigma_vasicek,strike_cap,a_vasicek,T_caplet,p_caplet)
price_caplet_ho_lee = fid.caplet_prices_ho_lee(sigma_ho_lee,strike_cap,T_caplet,p_caplet)
print(f"3ci - Caplet prices for T = 1,2,4,6,8,10 in Vasicek: {[np.round(10000*price_caplet_vasicek[2],5),np.round(10000*price_caplet_vasicek[4],5),np.round(10000*price_caplet_vasicek[8],5),np.round(
print(f"3ci - Caplet prices for T = 1,2,4,6,8,10 in Ho-Lee: {[np.round(10000*price_caplet_ho_lee[2],5),np.round(10000*price_caplet_ho_lee[4],5),np.round(10000*price_caplet_ho_lee[8],5),np.round(1000
price_cap_vasicek = sum(price_caplet_vasicek)
price_cap_ho_lee = sum(price_caplet_ho_lee)
print(f"3ca - Cap price Vasicek: {price_cap_vasicek*10000} bps, Cap price Ho-Lee: {price_cap_ho_lee*10000} bps")

def floorlet_prices_vasicek(sigma,strike,a,T,p):
    price_floorlet = np.zeros([len(T)])
    if type(strike) == int or type(strike) == float or type(strike) == np.int32 or type(strike) == np.int64 or type(strike) == np.float64:
        for i in range(2,len(T)):
            price_floorlet[i] = (1 + (T[i]-T[i-1])*strike)*fid.euro_option_price_vasicek(1/(1 + (T[i]-T[i-1])*strike),T[i-1],T[i],p[i-1],p[i],a,sigma,type = "call")
    elif type(strike) == tuple or type(strike) == list or type(strike) == np.ndarray:
        for i in range(2,len(T)):
            price_floorlet[i] = (1 + (T[i]-T[i-1])*strike[i])*fid.euro_option_price_vasicek(1/(1 + (T[i]-T[i-1])*strike[i]),T[i-1],T[i],p[i-1],p[i],a,sigma,type = "call")
    return price_floorlet

def floorlet_prices_ho_lee(sigma,strike,T,p):
    price_floorlet = np.zeros([len(T)])
    if type(strike) == int or type(strike) == float or type(strike) == np.int32 or type(strike) == np.int64 or type(strike) == np.float64:
        for i in range(2,len(T)):
            price_floorlet[i] = (1 + (T[i]-T[i-1])*strike)*fid.euro_option_price_ho_lee(1/(1 + (T[i]-T[i-1])*strike),T[i-1],T[i],p[i-1],p[i],sigma,type = "call")
    elif type(strike) == tuple or type(strike) == list or type(strike) == np.ndarray:
        for i in range(2,len(T)):
            price_floorlet[i] = (1 + (T[i]-T[i-1])*strike[i])*fid.euro_option_price_ho_lee(1/(1 + (T[i]-T[i-1])*strike[i]),T[i-1],T[i],p[i-1],p[i],sigma,type = "call")
    return price_floorlet

price_floorlet_vasicek = floorlet_prices_vasicek(sigma_vasicek,strike_floor,a_vasicek,T_caplet,p_caplet)
price_floorlet_ho_lee = floorlet_prices_ho_lee(sigma_ho_lee,strike_floor,T_caplet,p_caplet)
print(f"3ciii - Floorlet prices for T = 1,2,4,6,8,10 in Vasicek: {[np.round(10000*price_floorlet_vasicek[2],5),np.round(10000*price_floorlet_vasicek[4],5),np.round(10000*price_floorlet_vasicek[8],5)
print(f"3ciii - Floorlet prices for T = 1,2,4,6,8,10 in Ho-Lee: {[np.round(10000*price_floorlet_ho_lee[2],5),np.round(10000*price_floorlet_ho_lee[4],5),np.round(10000*price_floorlet_ho_lee[8],5),np.r
price_floor_vasicek = sum(price_floorlet_vasicek)
price_floor_ho_lee = sum(price_floorlet_ho_lee)
print(f"3ciii - Floor price Vasicek: {price_floor_vasicek*10000} bps, Cap price Ho-Lee: {price_floor_ho_lee*10000} bps")

# Problem 3d
def floor_vasicek_obj(strike,price_floor,sigma,a,T,p):
    price_floorlet = floorlet_prices_vasicek(sigma,strike[0],a,T,p)
    price_floor_new = sum(price_floorlet)
    sse = (price_floor - price_floor_new)**2
    return sse

def floor_ho_lee_obj(strike,price_floor,sigma,T,p):
    price_floorlet = floorlet_prices_ho_lee(sigma,strike[0],T,p)
    price_floor_new = sum(price_floorlet)
    sse = (price_floor - price_floor_new)**2
    return sse

param_0 = 0.05
result = minimize(floor_vasicek_obj,param_0,method = 'nelder-mead',args = (price_cap_vasicek,sigma_vasicek,a_vasicek,T_caplet,p_caplet),options={'xatol': 1e-20,'disp': False})
strike_floor_vasicek_client = result.x[0]
result = minimize(floor_ho_lee_obj,param_0,method = 'nelder-mead',args = (price_cap_ho_lee,sigma_ho_lee,T_caplet,p_caplet),options={'xatol': 1e-20,'disp': False})
strike_floor_ho_lee_client = result.x[0]
price_floorlet_vasicek_client = floorlet_prices_vasicek(sigma_vasicek,strike_floor_vasicek_client,a_vasicek,T_caplet,p_caplet)
price_floorlet_ho_lee_client = floorlet_prices_ho_lee(sigma_ho_lee,strike_floor_ho_lee_client,T_caplet,p_caplet)
price_floor_vasicek_client = sum(price_floorlet_vasicek_client)
price_floor_ho_lee_client = sum(price_floorlet_ho_lee_client)
print(f"3d - strike_floor_vasicek_client: {strike_floor_vasicek_client}, price_floor_vasicek_client: {10000*price_floor_vasicek_client} bps, strike_floor_ho_lee_client: {strike_floor_ho_lee_client},

# Problem 4
T_n, T_N = 5, 10
beta = 0.58
param_0 = 0.032, 0.6,-0.20
seed_sabr = 8
R_swaption, S_swaption = fid.swap_rate_from_zcb_prices(0,T_n,T_N,"annual",T_inter,p_inter)
print(f"4a - ATMF 5Y75 par swap rate: {R_swaption}")
N_swaption = len(K_swaption_offset)
```

```
K_swaption = K_swaption_offset/10000 + R_swaption*np.ones(N_swaption)
result = minimize(fid.fit_sabr_no_beta_obj,param_0,method = 'nelder-mead',args = (beta,iv_swaption_market,K_swaption,T_n,R_swaption),options={'xatol': 1e-8,'disp': False})
sigma_0, upsilon, rho = result.x
print(f"4b - Parameters from the SABR fit where beta: {np.round(beta,5)} are: sigma_0: {np.round(sigma_0,5)}, upsilon: {np.round(upsilon,5)}, rho: {np.round(rho,5)}, SSE: {result.fun}")
iv_fit = np.zeros([N_swaption])
for i in range(0,N_swaption):
    iv_fit[i] = fid.sigma_sabr(K_swaption[i],T_n,R_swaption,sigma_0,beta,upsilon,rho)

def sabr_simul(F_0,sigma_0,beta,upsilon,rho,M,T,method = "euler"):
    sigma, F = np.zeros([M+1]), np.zeros([M+1])
    sigma[0], F[0] = sigma_0, F_0
    delta = T/M
    Z = np.random.standard_normal([2,M])
    delta_sqrt = np.sqrt(delta)
    rho_sqrt_help = np.sqrt(1-rho**2)
    if method == "euler":
        for m in range(1,M+1):
            F[m] = F[m-1] + sigma[m-1]*F[m-1]**beta*delta_sqrt*Z[0,m-1]
            if F[m] <= 0:
                F[m] = F[m-1]
            sigma[m] = sigma[m-1] + upsilon*sigma[m-1]*delta_sqrt*(rho*Z[0,m-1] + rho_sqrt_help*Z[1,m-1])
    elif method == "milstein":
        for m in range(1,M+1):
            F[m] = F[m-1] + sigma[m-1]*F[m-1]**beta*delta_sqrt*Z[0,m-1] + 0.5*beta*sigma[m-1]**2*F[m-1]**(2*beta-1)*delta*(Z[0,m-1]**2-1)
            if F[m] <= 0:
                F[m] = F[m-1]
            Z3 = rho*Z[0,m-1] + rho_sqrt_help*Z[1,m-1]
            sigma[m] = sigma[m-1] + upsilon*sigma[m-1]*delta_sqrt*Z3 + 0.5*upsilon**2*sigma[m-1]*delta*(Z3**2-1)
    return F, sigma

T_simul_sabr, M_simul_sabr = T_n, 5000
seeds = [7,17,21,20]
for i, seed_sabr in enumerate(seeds):
    np.random.seed(seed_sabr)
    F_simul, sigma_simul = sabr_simul(R_swaption,sigma_0,beta,upsilon,rho,M_simul_sabr,T_simul_sabr,method = "milstein")
    t_simul_sabr = np.array([i*T_simul_sabr/M_simul_sabr for i in range(0,M_simul_sabr+1)])
    fig = plt.figure(constrained_layout=False, dpi = 300, figsize = (5,3))
    fig.suptitle(f"Simulated Forward rate and volatility in the SABR model, seed = {seed_sabr}", 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 = [0,1,2,3,4,5]
    ax.set_xticks(xticks)
    ax.set_xticklabels(xticks,fontsize = 6)
    ax.set_xlim([xticks[0]-0.02,xticks[-1]+0.02])
    plt.xlabel(f"Years",fontsize = 7)
    ax.set_yticks([0,0.02,0.04,0.06,0.08,0.1])
    ax.set_yticklabels([0,0.02,0.04,0.06,0.08,0.1],fontsize = 6)
    ax.set_ylim([0,0.101])
    plt.grid(axis = 'y', which='major', color=(0.7,0.7,0.7,0), linestyle='--')
    ax.set_ylabel(f"Forward swap rate",fontsize = 7)
    p1 = ax.scatter(t_simul_sabr, F_simul, s = 1, color = 'black', marker = ".",label="Forward swap rate")
    ax2 = ax.twinx()
    if i == 0 or i == 1:
        ax2.set_yticks([0,0.02,0.04,0.06,0.08,0.1])
        ax2.set_yticklabels([0,0.02,0.04,0.06,0.08,0.1],fontsize = 6)
        ax2.set_ylim([0,0.101])
    else:
        ax2.set_yticks([0,0.04,0.08,0.12,0.16,0.2])
        ax2.set_yticklabels([0,0.04,0.08,0.12,0.16,0.2],fontsize = 6)
        ax2.set_ylim([0,0.202])
    ax2.set_ylabel(f"Volatility",fontsize = 7)
    p2 = ax2.scatter(t_simul_sabr, sigma_simul, s = 1, color = 'red', marker = ".",label="Volatility")
    plots = [p1,p2]
    labels = [item.get_label() for item in plots]
    ax.legend(plots,labels,loc="upper right",fontsize = 5)

np.random.seed(2025)
N_simul_lookback, T_lookback = 5000, T_n
strike_lookback = R_swaption
payoff_lookback, price_lookback = np.zeros(N_simul_lookback), np.zeros(N_simul_lookback)
F_simul_lookback, sigma_simul_lookback = np.zeros([M_simul_sabr]), np.zeros([M_simul_sabr])
p_lookback = fid.for_values_in_list_find_value_return_value(T_n,T_inter,p_inter)
print(f"4d - Discount factor used to price the lookback option: {p_lookback}")
for n in range(0,N_simul_lookback):
    F_simul_lookback, sigma_simul_lookback = fid.sabr_simul(R_swaption, sigma_0, beta, upsilon, rho, M_simul_sabr, T_lookback)
    minimum = min(F_simul_lookback)
    payoff_lookback[n] = 10000*(strike_lookback-minimum)
    price_lookback[n] = p_lookback*sum(payoff_lookback[0:n+1])/(n+1)
print(f"4d - Price of the lookback option: {price_lookback[-1]}")

# PLot of the fitted ZCB term structures of spot and forward rates
fig = plt.figure(constrained_layout=False, dpi = 300, figsize = (5,3))
fig.suptitle(f"Calibrated Zero Coupon Spot and Forward Rates", 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 = [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([0,0.02,0.04,0.06,0.08])
ax.set_yticklabels([0,0.02,0.04,0.06,0.08],fontsize = 6)
ax.set_ylim([0,0.0825])
plt.grid(axis = 'y', which='major', color=(0.7,0.7,0.7,0), linestyle='--')
p1 = ax.scatter(T_inter[0:], R_inter[0:], s = 1, color = 'black', marker = ".",label="Spot rates")
p2 = ax.scatter(T_inter[0:], f_inter[0:], s = 1, color = 'red', marker = ".",label="forward rates")
p3 = ax.scatter(T_6M, L_6M, s = 1, color = 'green', marker = ".",label="6M forward Libor rates")
plots = [p1,p2,p3]
labels = [item.get_label() for item in plots]
ax.legend(plots,labels,loc="upper right",fontsize = 6)
bbox = {"facecolor": (1,1,1,0.8),"edgecolor": (0.7,0.7,0.7,0.5),"boxstyle": "Round"}
if interpolation_options["method"] == "hermite":
    ax.text(0.32,0.0032,f" method: {interpolation_options['method']} \n degree: {interpolation_options['degree']} \n transition: {interpolation_options['transition']}", fontsize = 6,linespacing = 1.
```

```
else:
    ax.text(0.32,0.0032,f" method: {interpolation_options['method']} \n transition: {interpolation_options['transition']}", fontsize = 6,linespacing = 1.7, bbox = bbox)
plt.show()

# Plot of Calibrated ZCB Spot Rates Fitted Using Vasicek
fig = plt.figure(constrained_layout=False, dpi = 300, figsize = (5,3))
fig.suptitle(f"Calibrated ZCB Spot Rates Fitted Using Vasicek", 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 = [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([0,0.02,0.04,0.06,0.08])
ax.set_yticklabels([0,0.02,0.04,0.06,0.08],fontsize = 6)
ax.set_ylim([0,0.0825])
plt.grid(axis = 'y', which='major', color=(0.7,0.7,0.7,0), linestyle='--')
p1 = ax.scatter(T_inter, R_inter, s = 1, color = 'black', marker = ".",label="Spot Rates")
p2 = ax.scatter(T_inter, R_vasicek, s = 1, color = 'red', marker = ".",label="Vasicek Fitted Spot Rates")
plots = [p1,p2]
labels = [item.get_label() for item in plots]
ax.legend(plots,labels,loc="upper right",fontsize = 6)
bbox = {"facecolor": (1,1,1,0.8),"edgecolor": (0.7,0.7,0.7,0.5),"boxstyle": "Round"}
ax.text(0.32,0.0032,f" r_0: {r0_vasicek} \n a: {a_vasicek} \n b: {b_vasicek} \n sigma: {sigma_vasicek}", fontsize = 6,linespacing = 1.7, bbox = bbox)
plt.show()

# Plot of Theta(t) in the HO-Lee model
fig = plt.figure(constrained_layout=False, dpi = 300, figsize = (5,3))
fig.suptitle(f"Theta in the Ho-Lee model", 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 = [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([-0.02,0,0.02,0.04,0.06,0.08])
ax.set_yticklabels([-0.02,0,0.02,0.04,0.06,0.08],fontsize = 6)
ax.set_ylim([-0.02,0.0825])
plt.grid(axis = 'y', which='major', color=(0.7,0.7,0.7,0), linestyle='--')
p1 = ax.scatter(T_star, theta, s = 1, color = 'black', marker = ".",label="Theta")
p2 = ax.scatter(T_inter[0:], f_inter[0:], s = 1, color = 'red', marker = ".",label="forward rates")
plots = [p1,p2]
labels = [item.get_label() for item in plots]
ax.legend(plots,labels,loc="upper right",fontsize = 6)
plt.show()

# PLot of simulated short rates in the Vasicek model
fig = plt.figure(constrained_layout=False,dpi=300,figsize=(5,3))
fig.suptitle(f"Simulated short rates in the Vasicek model",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 = [0,2,4,6,8,10]
ax.set_xticks(xticks)
ax.set_xticklabels(xticks,fontsize = 6)
ax.set_xlim([xticks[0]-0.01,xticks[-1]+0.01])
plt.xlabel(f"Time",fontsize = 7)
yticks1 = [0,0.02,0.04,0.06,0.08,0.1]
ax.set_yticks(yticks1)
ax.set_yticklabels(yticks1,fontsize = 6)
ax.set_ylim([yticks1[0],yticks1[-1] + (yticks1[-1]-yticks1[0])*0.02])
plt.grid(axis = 'y', which='major', color=(0.7,0.7,0.7,0), linestyle='--')
ax.set_ylabel(f"Short rate",fontsize = 7)
p1 = ax.scatter(t_simul, r_vasicek, s = 1, color = 'black', marker = ".",label="Simulated short rate")
p2 = ax.scatter(t_simul, mean_vasicek, s = 1, color = 'blue', marker = ".",label=f"Mean")
p3 = ax.scatter(t_simul, lb_vasicek, s = 1, color = 'red', marker = ".",label=f"Lower {size_ci} CB")
p4 = ax.scatter(t_simul, ub_vasicek, s = 1, color = 'red', marker = ".",label=f"Upper {size_ci} CB")
plots = [p1,p2,p3,p4]
labels = [item.get_label() for item in plots]
ax.legend(plots,labels,loc="upper right",fontsize = 5)
plt.show()

# PLot of simulated short rates in the Vasicek model
fig = plt.figure(constrained_layout=False,dpi=300,figsize=(5,3))
fig.suptitle(f"Simulated short rates in the Ho-Lee model",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 = [0,2,4,6,8,10]
ax.set_xticks(xticks)
ax.set_xticklabels(xticks,fontsize = 6)
ax.set_xlim([xticks[0]-0.01,xticks[-1]+0.01])
plt.xlabel(f"Time",fontsize = 7)
yticks1 = [0,0.02,0.04,0.06,0.08,0.1]
ax.set_yticks(yticks1)
ax.set_yticklabels(yticks1,fontsize = 6)
ax.set_ylim([yticks1[0],yticks1[-1] + (yticks1[-1]-yticks1[0])*0.02])
plt.grid(axis = 'y', which='major', color=(0.7,0.7,0.7,0), linestyle='--')
ax.set_ylabel(f"Short rate",fontsize = 7)
p1 = ax.scatter(t_simul, r_ho_lee, s = 1, color = 'black', marker = ".",label="Simulated short rate")
p2 = ax.scatter(t_simul, mean_ho_lee, s = 1, color = 'blue', marker = ".",label=f"Mean")
p3 = ax.scatter(t_simul, lb_ho_lee, s = 1, color = 'red', marker = ".",label=f"Lower {size_ci} CB")
p4 = ax.scatter(t_simul, ub_ho_lee, s = 1, color = 'red', marker = ".",label=f"Upper {size_ci} CB")
plots = [p1,p2,p3,p4]
labels = [item.get_label() for item in plots]
ax.legend(plots,labels,loc="upper right",fontsize = 5)
plt.show()

# Plot of swaption market implied volatilities
fig = plt.figure(constrained_layout=False, dpi = 300, figsize = (5,3))
fig.suptitle(f"Market and Fitted Swaption 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([0,0.05,0.1,0.15,0.2,0.25,0.3])
ax.set_yticklabels([0,0.05,0.1,0.15,0.2,0.25,0.3],fontsize = 6)
ax.set_ylim([0,0.305])
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_swaption_market, s = 6, color = 'black', marker = ".",label="IV market")
p2 = ax.scatter(K_swaption_offset, iv_fit, s = 2, 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)
plt.show()

fig = plt.figure(constrained_layout=False, dpi = 300, figsize = (5,3))
fig.suptitle(f"Price of the lookback payer swaption", 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 = [0,int((1/5)*N_simul_lookback),int((2/5)*N_simul_lookback),int((3/5)*N_simul_lookback),int((4/5)*N_simul_lookback),N_simul_lookback]
ax.set_xticks(xticks)
ax.set_xticklabels(xticks,fontsize = 6)
ax.set_xlim([xticks[0]-int(N_simul_lookback/50),xticks[-1]+int(N_simul_lookback/50)])
plt.xlabel(f"Number of simulations",fontsize = 7)
ax.set_yticks([0,50,100,150])
ax.set_yticklabels([0,50,100,150],fontsize = 6)
ax.set_ylim([0,152])
plt.grid(axis = 'y', which='major', color=(0.7,0.7,0.7,0), linestyle='--')
ax.set_ylabel(f"Price",fontsize = 7)
p1 = ax.scatter(np.array([i for i in range(1,N_simul_lookback+1)]), price_lookback, s = 1, color = 'black', marker = ".",label="Price by simulation")
p2 = ax.scatter(np.array([i for i in range(1,N_simul_lookback+1)]), np.ones(N_simul_lookback)*price_lookback[-1], s = 1, color = 'red', marker = ".",label=f"Final price: {price_lookback[-
1]}")
plots = [p1,p2]
labels = [item.get_label() for item in plots]
ax.legend(plots,labels,loc="upper right",fontsize = 5)
plt.show()
```