Introduction to Interest Rates, Bonds, Swaps and Swaptions

Theory, Code and Applications

Bailey Arm

September 21, 2023

Contents

1 Interest Rates

1.1 What are Interest Rates?

Interest rates are essentially the cost of borrowing money over a fixed period of time. For example, say you need $100 for a new coat and I am happy to lend you that $100, in exchange for you giving me 1% interest extra back. So, you get to buy a coat and I make a $1 profit after giving you the money. Intuitively, non-zero interest rates make sense; I am giving up the opportunity to use my $100 over the time that you are borrowing it from me.

1.2 Interest Rate Mathematics

1.2.1 Day Count Conventions

Day count conventions are the rules for calculating the amount of interest payable/accrued over a time period. The day count adjusted interest payable is:

$$\text{Interest Payable} = \text{Notional} \times \text{Day Count Factor} \times \text{Interest Rate Per Annum}$$

Here is an example of day count adjusting a $1mn loan at an annual interest rate of 5% between the days of Jan 1^{st} 2023 to Mar 1^{st} 2023:

Day Convention	Day Count Factor	Interest Payable on Loan
1/1	1/1	$\$1mn \times 1 \times 0.05 = \$50000/$
ACT/ACT	$(30 + 28\ )/365 = 0.1589$	$\$1mn \times 0.1589 \times 0.05 = \$7945.2$
ACT/365F	$(30 + 28\ )/365 = 0.1589$[1]	$\$1mn \times 0.1589 \times 0.05 = \$7945.2$
30/360	$(30 + 30)/360 = 0.16667$	$\$1mn \times 0.16667 \times 0.05 = \$8333.50$

We will be utilising day count conventions in later chapters, for example when evaluating the PV of an interest rate swap, day count conventions play a significant role.

Annualisation of Interest Rates

Say we have an annual rate of 10%. Dividing this in half, as we will do later on, does not give a semi-annual rate[2]. For example, a 10% annual interest on $1mn would be $100,000, whereas 5% semi-annually would be $1.05^2 \times \$1mn = \$102,500$ as we earn 5% on the initial $1mn principal

[2] This is to say that when compounding, one must take care with how they scale rates

and then both 5% on the principal again and also 5% on our original 5%. To find an equivalent semi-annual yield to an annual yield of 10%, we are essentially solving:

$$(1 + r)^2 = 1.10$$

So then we find:

$$r = \sqrt{1.10} - 1 = 0.0489$$

So then 10% compounded annually is equivalent to 4.9% compounded semi-annually.

To solve generally, one has that:

$$(1 + r)^m = 1 + R$$

Where r is the period rate, m is the number of periods in a year and R is the annual rate.

1.2.2 Present Value and Future Value of Cash Flows

Say that I have $X and that the current interest rates are r (for example $r = 0.04$), one might ask 'what is the future value of this principal?'. We can simply compute:

$$FV = (1 + r)X$$

So say I have $1000 and that current annual interest rates are 5%, the future value of my original principal will be $FV = 1.05 \times 1000 = 1050$. Now, what about after two years? We can simply multiply again by the same factor and get the answer:

$$FV = 1.05 \times (1.05 \times 1000) = 1102.5$$

Where we see a bigger increase as we are earning 'interest on interest'.

Now say that you want to give me $1000 one year in the future, with current interest rates at 5%. The most suitable question to ask would be 'what is that worth currently?'. This is equivalent to asking 'with current interest rates at 5%, what initial principal do I need to receive $1000 one year in the future?'.

Let X be the initial principal. We can formalise the question by solving for X in:

$$1.05 \times X = 1000$$

Simply dividing yields:

$$X = \frac{1}{1.05} 1000 = 952.38$$

Thus, the *present value* of $1000 dollars one year in the future under a 5% interest rate environment would be $952.38 [3]

Now imagine a more complex but realistic situation where interest rates are not stationary. An example of this can be seen in 1.1 This graph shows the Effective Federal Funds Rate (EFFR), a volume-weighted median of overnight federal funds transactions reported. This is a major

[3] Think of this as the fair cost to buy into this agreement to receive this cash flow. It is important to note that we are assuming the cash flow to be certain, with zero default probability

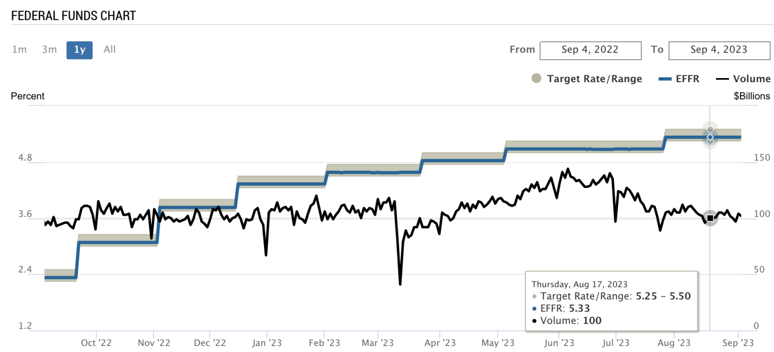

Figure 1.1: Effective Federal Funds Rate from 4th Sep 2022 to 4th Sep 2023

benchmark in the interest rate world. We can see that the Federal Reserve (the Fed) have been hiking (raising) interest rates in order to combat inflation, with EFFR increasing from 2.4% to more than 5.3%.

Let us start with a simple example. Let's say that interest rates are projected to follow 5% → 5.5% → 6% over the next two years, and say that I want to know how much my \$100 will be worth at the end of the two years.

I can simply compute this as such:

$$\$100 \times 1.055 \times 1.06 = \$111.83$$

Mathematically, let us say we have a series of projected interest rates $\mathbf{r} := (r_i)_{i=0}^{n}$ where r_i is the interest rate i years from now, and r_0 is the current interest rate. Say that I want to know what the present value of $\$X$ is $k \leq n$ years in the future is. I compute:

$$PV = X \prod_{i=1}^{k} (1 + r_i)$$

Let us further complicated this example by examining the present value of a series of cash flows $\mathbf{X} := (X_i)_{i=1}^{k}$, $k \leq n$. We compute the following sum in order to find the present value[4]:

$$PV = \sum_{i=1}^{k} \frac{X_i}{\prod_{j=1}^{i}(1 + r_i)}$$

Periodic Compounding

Some interest payments are calculated semi-annually or quarterly. To deal with this in present value calculations, we can adjust the annual rate in the following way:

[4]It is essential to point out that we are assuming that the payments are in the same currency, if they were not we could adjust using the fx rate, or use some chain of fx options to fix the conversion:

$$PV_{\text{FX CONVERTED}} = \sum_{i=1}^{k} \frac{X_i \times \text{fx}_i}{\prod_{j=1}^{i}(1 + r_i)}$$

Where fx_i is the fx conversion for our i^{th} payment

$$PV = \frac{X}{(1+i/m)^{n\times m}}$$

Where X is our periodic cash flow ($1000 annually may be seen as $250 quarterly), m is the number of periods in a year and n is the number of years.

Putting this into an example, say I have invested in company X and expect $1mn in 5 years. Furthermore, say that I am assuming that interest rates will stay static over this 5 year period at 4% compounded semi-annually. The present value of this cash flow is:

$$PV = \frac{1000000}{(1+0.04/2)^{5\times 2}} = \$820,348.30$$

So in other words, a fair value of this cash flow, assuming static interest rates and complete certainty in the cash flow, would be $820,348.30[5]

1.2.3 Yield of an Investment

Imagine that you have bought into some security that promises a series of cash flows $\mathbf{X}$ that, under the current interest rate expectations, is currently worth C. The yield on this investment is defined to the value y such that:

$$C = \sum_{i=1}^{n} \frac{X_i}{(1+y)^i}$$

For example, let us say that I have paid $858.76 for a 4% Government bond maturing in 10 years. What is the yield? We can see that given our bond is trading at a discount[6]. we know that the yield must be higher than the coupon rate. Guessing the yield to be 6%, gives a PV of $851.23, and thus the yield must be slightly lower. In fact, the exact yield is 5.889%.

There is in fact a formal way of doing this iteratively, using the Newton-Raphson method which I shall discuss in the later chapter 'Bond Yield Calculation under Newton-Raphson'

[5]This is technically incorrect, recall that the semi-annual rate is not exactly equal to half the annual rate

[6]Here are some terms used when discussing bonds:

(i) Discount Bond: The bond is trading below par. This happens when the coupon rate is below the required yield,

(ii) Bond is Trading at a Premium: The bond is trading above par. This is when the coupon rate is above the required yield

2 Interest Rate Modelling

This chapter can probably be skipped if you aren't interested in interest rate models or applying stochastic processes to finance - the rest of this document is not dependent on these models but I felt it best to include a chapter on modelling interest rates.

An assumption that I will build upon is that interest rates are mostly random[1] processes. It is perhaps better that I mention that the model that we will be building together will focus on short term movements (intra-week even) in interest rates.

2.1 The Most Basic Non-Deterministic Interest rate model

So, we could assume first that interest rates follow a random walk. Let r be the interest rate we want to model. Then, we could perhaps assume:

$$dr = dW_t$$

Where W_t is the standard 1-Dimensional Wiener process. This model is basically saying that at every interval, interest rates follow a normal distribution.

In Python:

```python
def simpleModel(initialRate, time, intervals):
    dt = time/intervals
    r = initialRate
    rates = []
    for i in range(intervals):
        r += (np.random.standard_normal() - np.random.standard_normal())/100
        rates.append(r)

    plt.figure(figsize = (12,6))
    plt.plot(rates)
    plt.xlabel(f'Time (1 = {dt} years)')
    plt.ylabel('Interest Rate')
    plt.title('Random Interest Rate Model')
    plt.show()
```

Running 'simpleModel(5, 1,1000)' for me, yields:

It is important to note that if you run this code, you *will* get a different graph, even when putting in the same variables. More accurately, we should scale the movements in line with a historical volatility of the interest rate:

$$dr = \sigma dW_t$$

[1] By mostly I am referring to the fact that interest rates will probably revert after some period of time, i.e. it is highly unlikely that the FED will continue to keep interest rates at around the 5% level

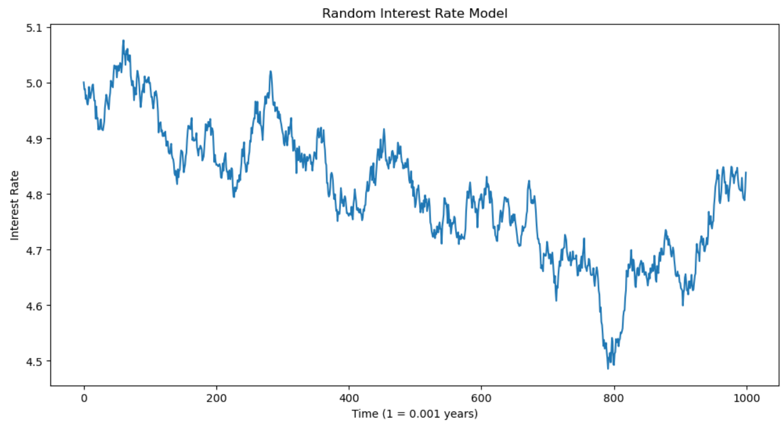

Figure 2.1: Simple Interest Rate Model, $dr = dW_t$

Where σ is the average historical volatility over our chosen time interval. Have a go at modifying my code above in order to include an extra input of historical volatility, and look at how this can affect the graph.

2.2 Mean-Reverting Interest Rates

One could safely assume that interest rates tend to return to some base level or a long term average. Let θ be this long term average. We could assume that each time step causes interest rates to creep back to this long term average with some strength κ. A possible model would be:

$$dr = \kappa(\theta - r)dt$$

Here is some code to plot the evolution of interest rates under this assumption:

```
def meanRevertingModel(kappa, longTermRate, initialRate, time, intervals
    ):
    dt = time/intervals
    r = initialRate
    rates = []

    for i in range(intervals):
        r0 = r
        r += kappa*(longTermRate - r0)*dt
        rates.append(r)

    plt.figure(figsize = (12,6))
    plt.plot(rates)
    plt.xlabel(f'Time (1 = {dt} years)')
    plt.ylabel('Interest Rate')
    plt.title('Mean-Reverting Interest Rate Model')
    plt.show()
```

Running this and 'meanRevertingModel(0.5,2,3, 1,1000)' shows why this model is not very realistic:

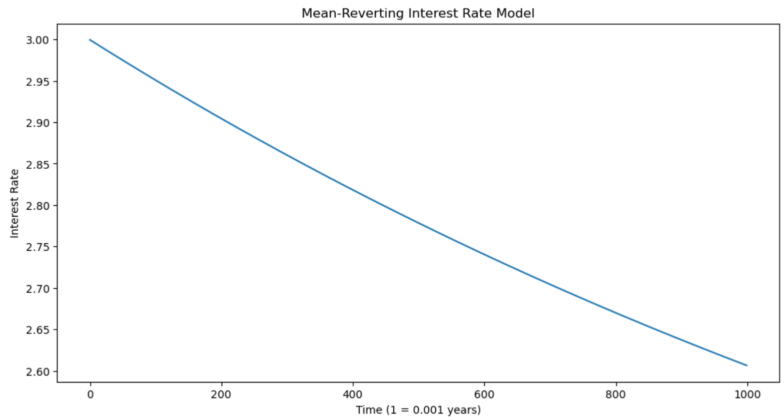

Figure 2.2: Mean-Reverting Model $dr = \kappa(\theta - r)dt$

It is clear to see why this model is never used on its own.

2.3 A Synthesis of Models

Now we combine the two models to make a more accurate model. We want to include some reversion characteristics but also randomness. We can simply add the models together:

$$dr = \kappa(\theta - r)dt + \sigma\sqrt{r}dW_t$$

Here is some Python code:

```
def interestRateModel(kappa, longTermAverage, initialRate, volatility, time, intervals):
    dt = time/intervals
    r = initialRate
    rates = []

    for i in range(intervals):
        r0 = r
        r += kappa*(longTermAverage - r0)*dt + volatility*np.sqrt(r0)*(np.random.standard_normal() - np.random.standard_normal())/100
        rates.append(r)

    plt.figure(figsize = (12,6))
    plt.plot(rates)
    plt.xlabel(f'Time (1 = {dt} years)')
    plt.ylabel('Interest Rate')
    plt.title('Mean-Reverting and Random Interest Rate Model')
    plt.show()
```

Running this and 'interestRateModel(0.3,2,5,0.2,1,1000)' shows the nature of the model (2.3).

It is clear to see why this synthesis model might be more suitable for interest rate modelling, not only does it include the random nature of interest rates, it also includes long term rate

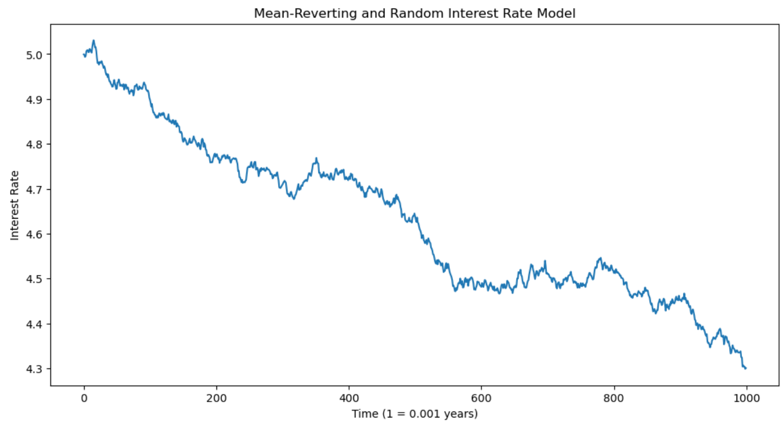

Figure 2.3: Synthesis Model, $dr = \kappa(\theta - r)dt + \sigma\sqrt{r}dW_t$

reversion. I have deliberately picked a large reversion number κ in order to demonstrate how interest rates behave under this model (note that as $\kappa \to 0$, we get a scaled version of our original model[2])

The paper 'Mean Reversion in Interest Rates: New Evidence from a Panel of OECD Countries' by Wu and Zhang could suggest that there are no statistically significant reversion characteristics in the interest rates of multiple different countries.

This chapter was just to show how one can build up from relatively simple assumptions and produce a model that exhibits interesting and hopefully useful results.

[2]Studies have found κ to be relatively low, close to 0 but skewed postively very slightly

3 Bonds

3.1 An Introduction to Bonds

Possibly the best example of using what we have learnt previously is in pricing bonds. Simply put, issuing a bond is an agreement to pay some coupon with some frequency until maturity and then a final coupon plus a 'par' at maturity. It can be thought of as a series of coupon payments at regular intervals (semi-annual for typical Government bonds and quarterly for some other bonds), plus some 'par' value at maturity.

For example, a 10 year 5% Government bond on a par of \$1000 has the following cash flows:

(i) $10 \times 2 = 20$ coupons of \$25[1] paid every 6 months for 10 years,

(ii) \$1000 at maturity (in 10 years)

3.1.1 Bond Pricing

To price this to yield a certain amount[2], we discount by our yield, say 3%:

$$PV = \sum_{i=1}^{20} \frac{\$25}{(1+0.03/2)^i} + \frac{\$1000}{(1+0.03/2)^{20}}$$

Noting that the left most part is actually a geometric series yields:

$$PV = \frac{\$25}{1.015} \frac{1-\frac{1}{1.015^{20}}}{1-\frac{1}{1.015}} + \frac{\$1000}{1.015^{20}} = \$25 \frac{1-\frac{1}{1.015^{20}}}{0.015} + \frac{\$1000}{1.015^{20}} = \$1171.69$$

It should be clear that we can substitute in algebraic expressions in place of our numbers in order to achieve a formulaic expression of the bond's PV:

$$PV = c \times \frac{1-\frac{1}{(1+y/k)^n}}{y/k} + \frac{\text{Par}}{(1+y/k)^n}$$

Where y is the annual yield expressed as a decimal, c is the periodic coupon payment and k is the number of payments per year (as stated earlier, this is typically 2 or 4)

[1] There are 5×2 coupons as the coupons are paid semi-annually. These coupons are \$25 each, simply as 5% of par must be paid each year, equating to \$50 per year or \$25 semi-annually

[2] This yield is usually chosen by comparing this bond to other bonds with equal maturity and equal credit quality

3.1.2 Finding the Yield on a Bond

Let's say that we have bought a 9% 5 year Government bond and its present value is \$1253.86. Finding the yield is an iterative process. If the yield was equal to the coupon rate (9%) then one should have that the present value of the bond is equal to its par of \$1000. As the PV is greater than par, we have that the coupon rate must be greater than the yield. Let's guess that the yield is 5%. The price is then:

$$PV = 0.09/2 \times \$1000 \frac{1 - \frac{1}{1.025^{5\times 2}}}{0.025} + \frac{\$1000}{(1.025)^{10}} = \$1175.04$$

Then we now have to adjust our 5% estimate down to say 3.4%, yielding a price of \$1255.51. The yield must be very close to 3.4%.

3.1.3 Zero Coupon Bonds

As their name suggests, Zero Coupon Bonds, or ZCBs are bonds which pay no coupon. Instead, they accrue interest over the entire maturity of the bond and pay it out at maturity. Simply put, ZCBs are priced at a steep discount to par. For example imagine a world where interest rates have sky rocketed to 10%. A \$1000 par 10 year ZCB would cost:

$$\frac{\$1000}{1.05^{20}} = \$376.89$$

3.1.4 Pricing a Bond between Coupon Payments

Sometimes when buying bonds, a price adjustment must be made between the day that you bought it and the last coupon date. Using the standard day count convention with US Treasuries of ACT/ACT, we can price a 5% 5 year bond bought on Mar 1^{st} with the next coupon on Sep 1^{st} with the required yield of 4.8%, understanding that there are 8 more coupons to be paid.

We start by finding the number of days between the settlement date and the next coupon under the ACT/ACT convention:

$$(30 + 30 + 31 + 30 + 31)/180 = 0.8444^{3}$$

Then, we price as follows:

$$PV = \frac{\$25}{1.024^{0.8444}} + \frac{\$25}{1.024^{0.8444}} \sum_{i=1}^{7} \frac{1}{1.024^{i}} + \frac{\$1000}{1.024^{0.8444+7}} = \$25 \frac{1 - \frac{1}{1.024^{7}}}{0.024 \times 1.024^{0.8444}} + \frac{\$1000}{1.024^{7.8444}}$$

Finding that $PV = \$986.42$

In fact, as we are buying between coupon payments, to buy the bond we must pay the above PV plus accrued interest, compensating the issuer for the amount they have earned but not received.

Accrued interest is calculated as:

[3] We divide by 180 here as the number of days between coupons is 180

$$\text{Accrued Interest} = \text{Coupon} \times \frac{\text{Number of Days from Last Coupon to Settlement Date}}{\text{Number of Days in a Coupon Period}}$$

In our case, the total bond price is: $\$986.42 + \$25 \times \frac{24}{180} = \$989.75$

Formula for Bond Price between Coupon Payments

$$PV = \text{Coupon} \times \frac{1 - \frac{1}{(1+i)^{n-1}}}{i \times (1+i)^{DCF}} + \frac{\text{Par}}{(1+i)^{n-1+DCF}}$$

Where:

$$\text{DCF} = \frac{\text{No. Days between Settlement Period and Next Coupon}}{\text{No. Days in a Coupon Period}}$$

Where these calculations are done in the relevant day count conventions

3.2 Bond Yield Calculation under Newton-Raphson

As mentioned previously, finding the yield requires a numerical process. We will be doing this through the Newton-Raphson method which states that if I want to find where some function $f(x)$ is equal to a, I iterate as follows:

$$x_{n+1} = x_n + \frac{a - f(x_n)}{\frac{df}{dx}(x_n)}$$

Writing the dirty price of a bond in terms of its yield, with $w = DCF$:

$$P_d(y) = \frac{c}{(1+y)^w}(w + \frac{1 - \frac{1}{(1+y)^{n-1}}}{y}) + \frac{P}{(1+y)^{n-1+w}}$$

We can follow my derivation:

This gives a formula that will iterate to find the semi-annual yield of a dirty bond expressed as a decimal.

Want to solve for the yield on the price of a dirty bond

Step 1: Write the dirty price as a function of yield:

$$P_d(y) = \frac{c}{(1+y)^w}\left(w + \frac{1-\frac{1}{(1+y)^{n-1}}}{y}\right) + \frac{P}{(1+y)^{n-1+w}}$$

Then we can use Newton-Raphson.

$$x_1 = x_0 + \frac{a - f(x_0)}{\left.\frac{df}{dx}\right|_{x=x_0}}$$

| Where x_l is our l^{th} yield approx, a is the dirty price, $f(x_l)$ is the dirty price under the x_l yield.

$$\frac{df}{dy} = \frac{d}{dy}\left(\underbrace{\frac{c}{(1+y)^w}\left(w + \frac{1-\frac{1}{(1+y)^{n-1}}}{y}\right)}_{1} + \underbrace{\frac{P}{(1+y)^{n-1+w}}}_{2}\right)$$

1

$$cw\frac{d}{dy}(1+y)^{-w} + c\frac{d}{dy}\left(y^{-w-1} - y^{-w-1}(1+y)^{n-1}\right) = -w^2c(1+y)^{-w-1} + c(-w-1)y^{-w-2} - c\left((-w-1)y^{w-2}(1+y)^{n-1} + (n-1)y^{-w-1}(1+y)^{-n-2}\right)$$

$$= \frac{-w^2c}{(1+y)^{w+1}} - \frac{c(w+1)(1-(1+y)^{n-1})}{y^{w+2}} - \frac{c(n-1)}{y^{w+1}(1+y)^{n+2}}$$

2

$$P\frac{d}{dy}(1+y)^{-(n-1+w)} = -P(n-1+w)(1+y)^{-n-w}$$

$$= \frac{-(n-1+w)P}{(1+y)^{n+w}}$$

So then:

$$y_{k+1} = y_k - \frac{P_d - \frac{c}{(1+y_k)^w}\left(w + \frac{1-\frac{1}{(1+y_k)^{n-1}}}{y_k}\right) - \frac{P}{(1+y_k)^{n-1+w}}}{c\left(\frac{w^2}{(1+y_k)^{n+1}} + \frac{(w+1)(1-(1+y_k)^{n-1})}{y_k^{w+2}} + \frac{n-1}{y_k^{w+1}(1+y_k)^{n+2}}\right) + \frac{(n-1+w)P}{(1+y_k)^{n+w}}}$$

Figure 3.1: Iterating to find the Bond Yield under Newton-Raphson

4 Interest Rate Swaps

4.1 What is an Interest Rate Swap?

An interest rate swap is simply an agreement (or a contract) between two parties to exchange cash flows on certain dates over a specified maturity. More specifically, one party agrees to pay some fixed percentage (say 5%) of some agreed amount called the notional ($\$1,000,000$ for example) in order to receive a variable, or *floating*, percentage that is bench marked by some index.

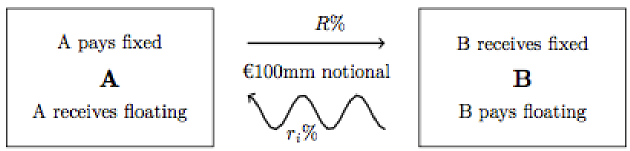

Figure 4.1: Interest Rate Swap Diagram

The image above is useful when understanding one payment during an IRS, however to think about the entirety of an IRS, imagine a sequence of these diagrams pasted above on another. For future ease, when I am referring to the payer of an IRS, I am simply referring to the payer of the fixed leg/receiver of the floating leg.

4.2 Pricing an Interest Rate Swap

Pricing an interest rate swap is fairly simple, from the perspective of the payer of a (vanilla) IRS, every period we pay some fixed amount and receive a variable amount.

Starting at the basics, imagine a one period IRS on a notional of N, with fixed rate f and floating rate r_0 in 6 months time. So, as the payer, in 6 months I will pay fN and receive rN, with my profit being the difference $N(r-f)$. Then I must discount this back in order to find the present value. So, the PV of the 1 period interest rate swap is $\frac{N(r-f)}{(1+k)}$ with k being the 6 month interest rate.

Now imagine I string these in a chain, receiving r_iN every 6 months and paying fN every 6 months. Then, my PV will be:

$$PV = N \sum_{i=1}^{n} \frac{(r_i - f)}{\prod_{j=1}^{i}(1+k_j)}$$

Perhaps putting this into an example could be useful. Say I am currently paying off my $\$1mn$ mortgage which is on a floating rate (let's call it the MB rate). Because of economic uncertainty and issues in the housing market, I believe that the floating rate will skyrocket from 6% to 20%

within a year. In order to do this, I will be the payer of a semi-IRS[1] on a notional of $\$1mn$, with a 1 year maturity receiving the MB rate and paying a fixed rate of 6%.

What I have essentially done here is cancelled the floating rate I pay on my mortgage, and will instead pay the fixed rate which is 6%. Now say that interest rates stay at 3% and the MB rate is expected to go from 6% → 11% → 15% (every 6 months). Then, the PV of the interest rate swap is:

$$\$1mn \times (\frac{11\% - 6\%}{1.03} + \frac{15\% - 6\%}{1.03^2}) = \$133,377$$

However say that the MB rate falls 6% → 5% → 2%. The PV of this would be $-\$47,412$.

4.2.1 Deciding the Fixed Rate on an Interest Rate Swap

Recall the price of an interest rate swap being:

$$PV = N \sum_{i=1}^{n} \frac{(r_i - f)}{\prod_{j=1}^{i}(1 + k_j)}$$

Typically when trading IRS, traders like to find the fixed rate at which the IRS has 0 value. In essence, we are solving:

$$N \sum_{i=1}^{n} \frac{(r_i - f)}{\prod_{j=1}^{i}(1 + k_j)} = 0$$

This is hopefully easy to solve:

$$\sum_{i=1}^{n} \frac{(r_i - f)}{\prod_{j=1}^{i}(1 + k_j)} = 0 \implies \sum_{i=1}^{n} \frac{r_i}{\prod_{j=1}^{i}(1 + k_j)} = \sum_{i=1}^{n} \frac{f}{\prod_{j=1}^{i}(1 + k_j)}$$

We have that:

$$\sum_{i=1}^{n} \frac{f}{\prod_{j=1}^{i}(1 + k_j)} = f \sum_{i=1}^{n} \frac{1}{\prod_{j=1}^{i}(1 + k_j)}$$

Yielding

$$f \sum_{i=1}^{n} \frac{1}{\prod_{j=1}^{i}(1 + k_j)} = \sum_{i=1}^{n} \frac{r_i}{\prod_{j=1}^{i}(1 + k_j)} \implies f = \frac{\sum_{i=1}^{n} \frac{r_i}{\prod_{j=1}^{i}(1+k_j)}}{\sum_{i=1}^{n} \frac{1}{\prod_{j=1}^{i}(1+k_j)}}$$

This is called the swap rate, or the mid-market rate. For example, say that we know for certain that the discounting interest rates k_i will remain at 4.5% for the next 3 years, and the market is pricing for the floating rate to go from 3% → 5% → 4%. We can now find the swap rate of a 3y IRS:

$$f = \frac{\frac{0.03}{1.045} + \frac{0.05}{1.045^2} + \frac{0.04}{1.045^3}}{\frac{1}{1.045} + \frac{1}{1.045^2} + \frac{1}{1.045^3}} = 0.0399 = 4\%$$

[1]The 'semi' prefix refers to the frequency of payments, i.e. every 6 months

In short, the swap rate contains useful information about the rate of return on an interest rate swap.

It is important to note that I have assumed that the interests given are quoted and scaled by their appropriate day count convention. Technically, the actual pricing formula, from the perspective of the payer is:

$$PV = N(-f\sum_{i=1}^{N_1}\frac{\text{dcf(fixed)}_i}{\prod_{k=1}^{i}(1+\alpha_k)} + \sum_{j=1}^{N_2}\frac{r_j\text{dcf(floating)}_j}{\prod_{k=1}^{j}(1+\beta_k)})$$

Where N_1, N_2 are the number of payments on the fixed and floating leg, respectively, $\text{dcf}(...)_i$ is the day count factor for the i^{th} payment of the ... leg, r_j is the floating rate on the j^{th} payment and α_k, β_k are the discounting rates for the fixed and floating leg respectively. Although this is a rather daunting formula, an example illustrates the utility of explicitly stating each involved factor.

For example, say I'm the payer of a \$100mn 1Y semi-quarterly IRS, with fixed rate of 2%. Furthermore, let's say that the fixed leg payments happen on the 1^{st} every 3 months and the floating is on the 1^{st} every 6 months, with our calculations happening under the 30/360 day count convention, and assuming that the discount rate is 5% per annum, we can value this swap as follows.

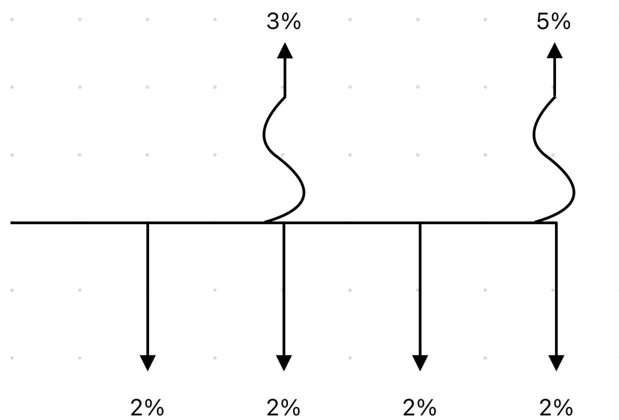

Figure 4.2: The 1Y Semi-Quarterly IRS

Fixed Leg

The fixed leg pricing formula is simply:

$$PV_{\text{Fixed Leg}} = N * f\sum_{i=1}^{N_1}\frac{\text{dcf(fixed)}_i}{\prod_{k=1}^{i}(1+\alpha_k)}$$

Where $N = 100mn$, $f = 0.02$, $\text{dcf(fixed)}_i = (30+30)/360 = 0.16667$ and $\alpha_k = (1+0.05)^{1/4} - 1 = 0.0123$, we have:

$$PV_{\text{Fixed Leg}} = 100mn * 0.02 * 0.1667\sum_{i=1}^{4}\frac{1}{1.0123^i} = \$1,293,579$$

Now to calculate the PV of the floating leg, under the assumption that the floating rate will rise to 3% and then 5%:

$$PV_{\text{Floating Leg}} = N(\frac{0.03 * (6 * 30)/360}{\sqrt{1 + 0.05}} + \frac{0.05 * (6 * 30)/360}{1.05}) = \$3,844,802$$

Then subtracting the fixed PV from the floating PV yields a price of $2,551,223. In short, adjusting the formula to include day count conventions allows us to be more flexible and to price different structures/IRS with different day count conventions on either leg.

4.3 Python Pricing of Interest Rate Swaps

I have included code to be used in pricing interest rate swaps in my post, however I felt a quick explanation necessary.

```
import numpy as np

def swapPrice(notional, floating, fixed, volatility, maturity, discountingRates):
#floating is an array of your expectations of the floating leg, say there are n entries,
#periods is the number of different coupon payments,
#discounting rates is an array of expecations of continuous discounting rates on each period (must also be n entries)
floatingLeg = 0
periods = len(floating)
for i in range(periods):
    floatingLeg += (notional*floating[i])*np.exp(-discountingRates[i]*(i+1)*maturity/periods)

fixedLeg = 0
for j in range(periods):
    fixedLeg += (notional*fixed)*np.exp(-discountingRates[i]*(i+1)*maturity/periods)

print(f'The price that the payer of this swap should pay is: {floatingLeg - fixedLeg}')
```

Let's say that I want to be the payer of an XYZ swap, thinking that the XYZ rate will go as follows 3% → 3.5% → 3.8% → 4.5% over the next year. Furthermore, say for some reason I want to hedge against this rise and want to be the payer of a swap on XYZ, maturing in a year. Obviously, I want to know how much this will cost. We can then simply run the following code (after running the code above):

```
swapPrice(1000000, [0.035, 0.038, 0.045], 0.03, 0.02, 1, [0.045, 0.043, 0.043])
```

Resulting in a price of $28,299. Now say instead that I pay that much and the rate will go: 3% → 4% → 5.2% → 6.5%. A good question to ask is, what is my profit? It is calculated through calculating:

```
swapPrice(1000000, [0.04, 0.052, 0.065], 0.03, 0.02, 1, [0.045,
    0.043, 0.043])
```

And then subtracting, resulting in a profit of \$65987 − \$28299 = \$37688[2]

4.4 Vanilla Basis Swaps

A basis swap is a swap in which both legs are floating. For example, say I want to buy a \$1mn 6mo 1M \$LIBOR vs 3M \$LIBOR, i.e. I want to pay the 1mo \$LIBOR rate for 6mo and receive the 3mo \$LIBOR rate over the same period of time. The question is how to price this. We will first start by pricing the leg to pay:

$$PV_{\text{Paying Leg}} = N\sum_{i=1}^{N_1}\frac{p_i \text{dcf}_i}{\prod_{k=1}^{i}(1+\alpha_k)}$$

Where p_i is the rate to pay at time i. Pricing the receiving leg is identical apart from the rate, dcf and discounting:

$$PV_{\text{Receiving Leg}} = N\sum_{j=1}^{N_2}\frac{r_j \text{dcf}_j}{\prod_{k=1}^{j}(1+\beta_k)}$$

Note that typically a spread adjustment is made to the shorter tenor rate such that the total PV is 0. Adding a spread onto the paying leg, we have that the PV of a basis swap is:

$$PV = N(-\sum_{i=1}^{N_1}\frac{(p_i+S)\text{dcf}_i}{\prod_{k=1}^{i}(1+\alpha_i)} + \sum_{j=1}^{N_2}\frac{r_j \text{dcf}_j}{\prod_{k=1}^{j}(1+\beta_k)})$$

It is a simple exercise to find the spread necessary for a zero PV:

$$PV = 0 \implies \sum_{i=1}^{N_1}\frac{(p_i+S)\text{dcf}_i}{\prod_{k=1}^{i}(1+\alpha_i)} = \sum_{j=1}^{N_2}\frac{r_j \text{dcf}_j}{\prod_{k=1}^{j}(1+\beta_k)}$$

Pulling out our adjusting spread S:

$$S\sum_{i=1}^{N_1}\frac{\text{dcf}_i}{\prod_{k=1}^{i}(1+\alpha_k)} + \sum_{i=1}^{N_1}\frac{p_i \text{dcf}_i}{\prod_{k=1}^{i}(1+\alpha_k)} = \sum_{j=1}^{N_2}\frac{r_j \text{dcf}_j}{\prod_{k=1}^{j}(1+\beta_k)}$$

In which we have that :

$$S = \frac{\sum_{j=1}^{N_2}\frac{r_j \text{dcf}_j}{\prod_{k=1}^{j}(1+\beta_k)} - \sum_{i=1}^{N_1}\frac{p_i \text{dcf}_i}{\prod_{k=1}^{i}(1+\alpha_k)}}{\sum_{i=1}^{N_1}\frac{\text{dcf}_i}{\prod_{k=1}^{i}(1+\alpha_k)}}$$

Going back to my example, let's say that the market is expecting the 1mo \$LIBOR rate to follow: 3%,4%,3.5%, 3.1%, 3%,3% and the 3mo \$LIBOR rate to follow: 3.7%, 3.6%. To keep

[2] It is important to note that I have placed assumptions of stability in the discounting rates, these can and do change and will affect the prices

this example nice and simple we will just assume that we can discount by an annual rate of 2%, and that we can use the 30/365 day count convention, along with every payment date falling on the 1^{st} of its month.

To find the PV of the paying leg, we use the formula:

$$PV_{\text{Paying Leg}} = N \sum_{i=1}^{N_1} \frac{p_i \text{dcf}_i}{\prod_{k=1}^{i}(1+\alpha_k)}$$

With $N = \$1mn$, $\alpha_k = 1.02^{1/12} - 1 = 0.00165$ and $\text{dcf}_i = 30/365 = 0.0822$. Our PV is then:

$$PV_{\text{Paying Leg}} = \$1mn * 0.0822 \sum_{i=1}^{6} \frac{p_i}{1.00165^i} = \$16,020.85$$

In a similar fashion, we find the PV of our receiving leg to be $35,472.12. So then, the PV of this basis swap would be $35,472.12 - $16,020.85 = $19,451.27.

Plugging our values in to the spread formula would yield a spread adjustment of 4bps[3]

4.5 Cross Currency Basis Swap

4.5.1 Non MTM (Mark-To-Market) Cross Currency Basis Swap

A cross currency basis swap (XCBS) is very similar to a standard basis swap, apart from the fact that in a XCBS, the legs are paid in different currencies. Take for example a $1mn 1y GBP/USD Semi-Quarterly XCBS on two imaginary rates - the pound rate and the dollar rate. On one side of this swap, I would be paying the pound rate in pounds every 6 months for 1y on a notional of $1mn, and receiving the dollar rate in dollars every 3 months for 1y on a notional of $1mn.

It is best to draw a diagram in order to understand this type of swap.

Where p_i is the pound rate of the i^{th} period and d_i is the dollar rate of the i^{th} period. It is important to note that in XCBSs, both day count conventions and discounting rates are different. We will price the example XCBS first, from the perspective of the payer of the pound, and then build up from there.

Technically the diagram I have drawn is incorrect. Before any interest is exchanged, the notionals are exchanged (for example a $1mn EUR/USD XCBS would involve one party giving the other $1mn in euros and the other party paying $1mn). At maturity of the swap, these notionals are converted back to their original currency and then switched back.

[3]Given the PV, it is relatively simple to find the spread adjustment:

$$S = \frac{\sum_{j=1}^{N_2} \frac{r_j \text{dcf}_j}{\prod_{k=1}^{j}(1+\beta_k)} - \sum_{i=1}^{N_1} \frac{p_i \text{dcf}_i}{\prod_{k=1}^{i}(1+\alpha_k)}}{\sum_{i=1}^{N_1} \frac{\text{dcf}_i}{\prod_{k=1}^{i}(1+\alpha_k)}} = \frac{PV/N}{\sum_{i=1}^{N_1} \frac{\text{dcf}_i}{\prod_{k=1}^{i}(1+\alpha_k)}}$$

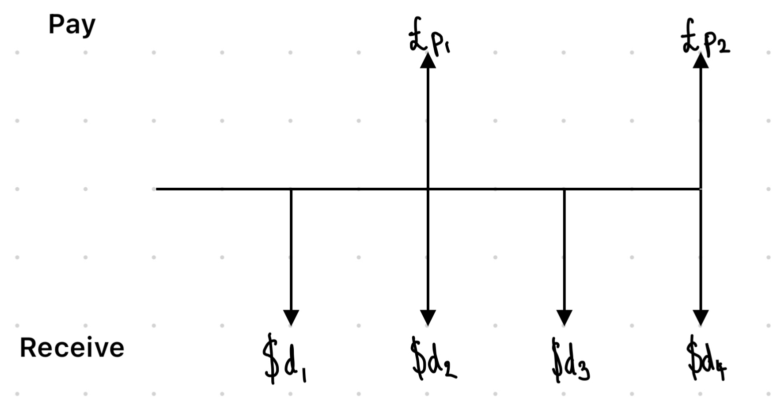

Figure 4.3: Cross Currency Basis Swap

Let us start pricing these models through an example - that of a €1mn EUR/USD 1y semi-semi non-MTM with a fixing rate of 1.3, i.e. €1 = \$1.3. We have the following cash flows:

(i) Pay €1mn and receive \$1.3mn (to/from counterparty),

(ii) Pay $(x_1 + Z)\%$ on \$1.3mn, receive $y_1\%$ on €1mn,

(iii) Pay $(x_2 + Z)\%$ on \$1.3mn, receive $y_2\%$ on €1mn,

(iv) Pay \$1.3mn and receive €1mn

Breaking this swap into parts (i), (ii)-(iii), (iv):

i - PV of Initial Notional Exchange

$$PV = F_0 N k_0 - N l_0$$

Where N is our notional amount, k_i is the discount rate on a USD cash flow, l_i is the discount rate on a EUR cash flow and F_0 is the initial fixing rate.

ii-iii - PV of Basis Section

$$PV = NF_0 \sum_{i=1}^{2} (x_i + Z)\text{dcf}(\$)_i k_i - N \sum_{j=1}^{2} y_j \text{dcf}(€)_j l_j$$

iv - PV of Final Currency Swap

$$PV = f_0 N l_2 - N F_0 k_2$$

So then we can add all of these to get the PV of a cross currency basis swap:

$$PV = (F_0 N k_0 - N l_0) + NF_0 \sum_{i=1}^{N_1} (x_i + Z)\text{dcf}(\$)_i k_i - N \sum_{j=1}^{N_2} y_j \text{dcf}(€)_j l_j + f_0 N l_{N_2} - N F_0 k_{N_1}$$

It is important to note that this is from the perspective of the *payer of the spread*. We can find the mid-market spread value such that the swap has 0 PV:

$$Z = \frac{F_0 k_0 - N l_0 + \sum_{j=1}^{N_2} y_j \mathrm{dcf}(€)_j l_j - F_0 \sum_{i=1}^{N_1} x_i \mathrm{dcf}(\$)_i k_i + f_0 l_{N_2} - F_0 k_{N_1}}{F_0 \sum_{i=1}^{N_1} \mathrm{dcf}(\$)_i k_i}$$

I will not be looking at these in much detail but there is another important type of swap - the mark-to-market XCBS. Pricing these is very similar to above, however we use that currencies change in value respective to one another, for example today \$1 might be £0.95 and could be £0.9 tomorrow.

$$\begin{aligned} PV = {} & N F_0 k_0 - N l_0 \\ & + N f_{j-1} \sum_{i=1}^{N_1} (x_i + Z) \mathrm{dcf}(\$)_i k_i - N F_0 \sum_{j=1}^{N_2} y_j \mathrm{dcf}(€)_j l_j \\ & + N \sum_{i=1}^{N_1} (f_{j-1} - f_j) \mathrm{dcf}(\$)_i k_i \\ & + N f_{N_1} k_{N_1} - N F_0 l_{N_2} \end{aligned}$$

As an exercise, send me the formula for the mid market rate of this swap!

5 An Introduction to Swaptions

5.1 What is a Swaption?

5.1.1 Definition of a Swaption

A swaption is an option to enter a determined leg of an IRS at a specified rate (called the strike rate) at expiry.

An example of a swaption is a 'GBP $1mn 6m 2y receiver 4%', i.e the option to receive fixed and pay floating on a 2y GB rate swap if, in 6 months, the rate is below 4% (think of this as receiver is a put and payer is a call).

5.1.2 Swaption Payoffs

Payer Swaption

A payer swaption is the right to enter into the payer side of a swap at a specified rate, called the strike rate. The payoff increases linearly with swap rate (if the swap rate is greater than the strike rate, in which case the payoff is 0).

Receiver Swaption

A receiver swaption is the right to enter into the receiver side of a swap at a strike rate. The option can be exercised if the strike rate is greater than the swap rate, with payoff increasing linearly as swap rate decreases towards 0 when strike rate equals swap rate.

Straddle

Akin to vanilla equity options, a swaption straddle is simply a combination of a receiver and a payer swaption with the same strike rate and maturity. Similar to vanilla options, our payoff is 0 when swap rate equals strike rate, but increases with deviation from strike rate.

6 Swaption Pricing Theory

During the rest of this document, I will be using ideas from vanilla equity option pricing theory to help gain intuition behind how different factors affect the price of a swaption. Arguably, the biggest factor in option pricing is volatility. Volatility is simply a measure of how much an underlying asset moves over a period of time; it is a quantification of uncertainty. In swaption pricing, this is no different. As volatility increases, so too does the probability that our swap rate is where we want it to be.

A common mistake is thinking that an increase in volatility also increases the chance that our swap rate is not where we want it to be and thus the swaption should be cheaper. This is false thinking; options have a one-sided payoff structure and thus we do not care about how far the swap rate goes away from us.

To put it further into context, imagine I have bought a 100 strike call on company XYZ, maturing in 1 year, and XYZ is currently trading at 110. At the time of purchase, implied volatility was at 20%, meaning that typically XYZ moves up or down 20% over a year. If by the end of the year, XYZ does fall 20% to 88, my payoff from the option will be 0. If instead it fell to 20, my payoff is still 0.

6.1 A Brief Aside - Option Intuition Advice

A useful bit of advice that I learnt during my internship was that when dealing with options and trying to understand the intuition behind almost anything, the best place to start is with the payoff structure. As an example, we can informally derive characteristics about the delta of a vanilla option through looking at the payoff and how that changes when our underlying moves around key points:

To intuit what the delta of an extremely deep ITM vanilla call option is, we start off with the payoff. Deeply ITM tells us that our strike is significantly below the underlying, i.e our payoff is $\max(S-K,0),\ K << S$ where S is the underlying price and K is the strike. Clearly, if we are super deep ITM, our payoff will approach the underlying S. The delta is then simply the rate of change of our payoff with respect to the underlying[1]. It is hopefully clear that $\frac{\partial}{\partial S}S = 1$ and therefore the delta of a deep ITM call is 1.

Now to intuit the delta of an ATM call option, we can do a similar trick. The payoff of an ATM call is $\max(S - S_0, 0)$ where S_0 is the strike when the option is bought. If the underlying moves up by 1, our payoff is 1 and if it moves down by 1, our payoff is 0. As a rough assumption, let us further assume that the probability a stock moves up equals the probability it moves down.

[1]It is important to note that delta is actually the rate of change of the price with respect to the underlying, but we can use payoff as a proxy

Then, our expected change in payoff will be $\frac{1}{2} \times 1 + \frac{1}{2} \times 0 = \frac{1}{2}$. Then we have found that the delta of an ATM vanilla call is 0.5.

I will leave it up to you to find and intuit the delta of a far OTM option (when $K >> S$).

6.2 Back to Swaption Pricing

Different to equity options, our implied volatility is measured in bps, or basis points. These are one one hundredth of a percentage point. As I mentioned before, implied volatility affects prices positively, and implied volatility increases with maturity[2]

6.2.1 Binary Pricing Model

This is the simplest model used to price options, and is perhaps best used in illustrative examples. In short, the binomial asset pricing model assumes that the underlying swap rate only moves up or down by a fixed amount. In the example below, I have drawn a diagram of an example of the model. We start off by determining the initial rate, say 2.50. Then, over each time interval

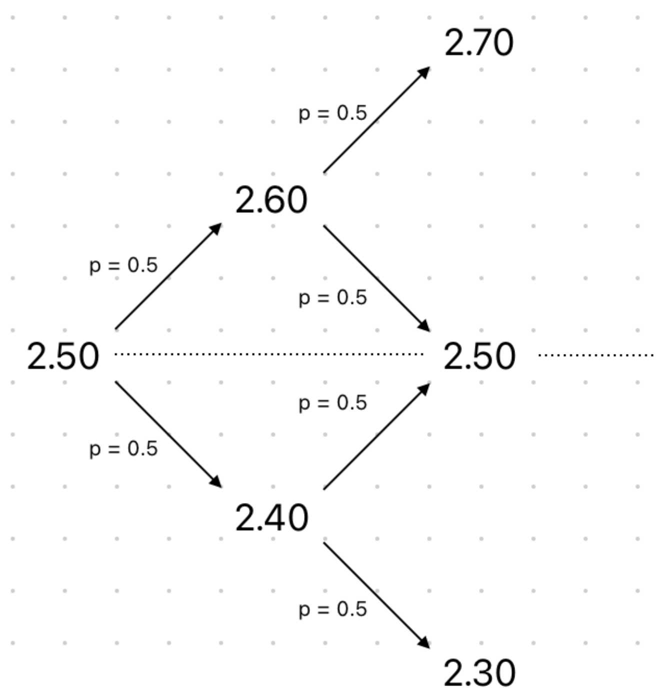

Figure 6.1: Two-Period Binomial Asset Pricing Model

(could be a day, hour, month), there is a 50% chance that the rate moves up by 10bps and the same chance it moves down 10bps. I won't be using this model much, but it is certainly a useful model to look at.

[2]Try to think as to why this is!

6.2.2 Pricing European Swaptions

European swaptions are simply swaptions that give the right to buy into a swap at, and only at, maturity.

The present value of a payer swaption is:

$$Ne^{-r\tau}(S\Phi(d_1) - K\Phi(d_2))$$

Where:

(i) N is the notional amount,

(ii) S is the forward swap rate,

(iii) K is the strike rate,

(iv) $d_1 = \frac{1}{\sigma\sqrt{\tau}}(\ln(\frac{S}{K}) + \frac{\sigma^2}{2}\tau)$,

(v) $d_2 = d_1 - \sigma\sqrt{\tau}$,

(vi) $\Phi(x) := \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-t^2/2}dt$,

(vii) r is some discounting rate,

(viii) τ is the maturity of the swaption

7 Swaption Greeks

7.1 Delta

7.1.1 Swaption Delta Derivation

This derivation can definitely be skipped, it does not give any intuition behind delta but instead focuses on deriving the formula. The delta of a swaption is the change in price with respect to a 1bps change in the floating rate. Formalising, it is:

$$\Delta = \partial_S N e^{-r\tau}(S\Phi(d_1) - K\Phi(d_2))$$

We will start the derivation by calculating $\partial_S \Phi(d_1)$.

$$\partial_S \Phi(d_1) = \frac{1}{\sqrt{2\pi}} \partial_S \int_{-\infty}^{d_1} e^{-t^2/2} dt = \frac{1}{\sqrt{2\pi}} e^{-d_1^2/2} \partial_S d_1$$

And with

$$\partial_S d_1 = \partial_S (\frac{1}{\sigma\sqrt{\tau}}(\ln(\frac{S}{K}) + \frac{\sigma^2}{2}\tau)) = \frac{1}{\sigma\sqrt{\tau}} \frac{1}{S}$$

So then, letting $\phi(x) = \Phi'(x)$, i.e. the normal pdf, one has that

$$\partial_S \Phi(d_1) = \phi(d_1) \frac{1}{S\sigma\sqrt{\tau}}$$

Noting that $\Phi(d_2) = \Phi(d_1 - \sigma\sqrt{\tau})$, and by the chain rule, we have:

$$\partial_S \Phi(d_2) = \phi(d_2) \partial_S (d_1 - \sigma\sqrt{\tau}) = \phi(d_2) \frac{1}{S\sigma\sqrt{\tau}}$$

Noting that by the product rule, delta is:

$$\Delta = N e^{-r\tau}(S\partial_S \Phi(d_1) + \Phi(d_1) + K\partial_S \Phi(d_2))$$

Noting that

$$\phi(d_2) = \frac{1}{\sqrt{2\pi}} e^{-(d_1^2 - 2d_1\sigma\sqrt{\tau} + \sigma^2\tau)/2} = \phi(d_1) e^{(\ln(\frac{S}{K}) - \sigma^2\tau/2)/\sigma\sqrt{\tau}) \times \sigma\sqrt{\tau} + \sigma^2\tau/2}$$

Which then shows

$$\phi(d_2) = \phi(d_1)\frac{S}{K}$$

So finally,

$$N e^{-r\tau}(\Phi(d_1) + \frac{1}{S\sigma\sqrt{\tau}} S\phi(d_1) - \frac{1}{S\sigma\sqrt{\tau}} K\phi(d_2)) = N e^{-r\tau}(\Phi(d_1) + \frac{1}{S\sigma\sqrt{\tau}}(S\phi(d_1) - K\frac{S}{K}\phi(d_1)) = N e^{-r\tau}\Phi(d_1)$$

This is probably recognisable from the delta of a vanilla equity option is $\Phi(d_1)$.

The delta of a receiver swaption is simply $Ne^{-r\tau}(\Phi(d_1) - 1)$

7.1.2 Delta Structure of Payer Swaption

It is a useful exercise to try to think about the delta structure as strike varies for a swaption, however I will include Python code here and in a Jupyter notebook so that you can play around and investigate for yourself.

Summarising the previous chapter, we found that the delta of a payer swaption is

$$\Delta = Ne^{-r\tau}\Phi(d_1)$$

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

def payerDelta(notional, swapRate, strike, volatility, maturity, discountingRate):
    d1 = (1/(volatility * np.sqrt(maturity)))*(np.log(swapRate/strike) + 0.5*volatility**2 * maturity)
    n1 = norm.cdf(d1, 0, 1)
    delta = notional*n1*np.exp(-discountingRate*maturity)
    return delta

def plot_deltaVSstrike(notional, swapRate, strike, volatility, maturity, discountingRate):
    k = np.linspace(0.8*strike, 1.2*strike, 1000)
    plt.figure(figsize = (12,6))
    plt.plot(k, payerDelta(notional, swapRate, k, volatility, maturity, discountingRate))
    plt.title('Delta-Strike Relationship')
    plt.xlabel('Strike Rate')
    plt.ylabel('Swaption Delta')
    plt.show()

plot_deltaVSstrike(1, 0.05, 0.05, 0.02, 1, 0.05)
```

Running this yields the following output:

So we can see that in 7.1 the delta is 1 for a deep ITM payer swaption, 0.5 for an ATM payer swaption and 0 for a far OTM payer swaption (exactly the same as an equity option).

7.1.3 Delta Hedging a Swaption

Say we have bought a swaption, the \$1mn 1y*5y payer struck ATM has a delta of \$1000. If the 1yf 5y rate rises by 1bp, our swaption appreciates by \$1000 and the inverse happens. Now say I want to have zero exposure to the underlying swap and instead want to profit from volatility. I can do this by buying the underlying swap in a size large enough such that my DV01 equals my delta. For instance let the \$1mn 5y payer have a DV01 of \$4000. Then I delta hedge by selling \$0.25mn 5y payer (or equally buying a \$0.25mn 5y receiver)

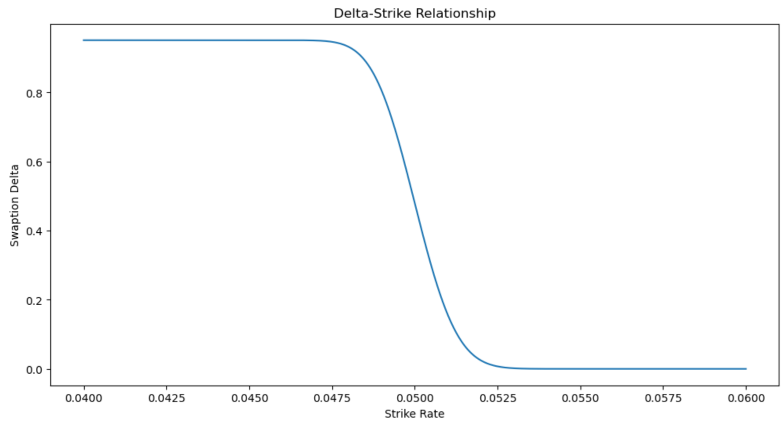

Figure 7.1: Delta vs Strike Rate for a Payer Swaption

7.2 Gamma

7.2.1 Swaption Gamma Derivation

Recall the delta of a payer swaption:

$$\Delta = Ne^{-r\tau}\Phi(d_1)$$

Gamma, or Γ is the rate of change of delta with respect to the swap rate. So, we want to compute:

$$\Gamma = Ne^{-r\tau}\partial_S\Phi(d_1) = Ne^{-r\tau}\phi(d_1)\partial_S d_1$$

We have already computed the majority of this:

$$\Gamma = \frac{Ne^{-r\tau}\phi(d_1)}{S\sigma\sqrt{\tau}}$$

Now let us theorise about what the gamma-strike relationship looks like. I can (very very roughly) guess, by looking at the delta-strike graph and knowing that gamma is going to be the gradient of the curve, that gamma is 0 when we are far away from the current swap rate, and high when close to the swap rate.

7.2.2 Swaption Gamma Python Code

Building code to plot gamma is not hard at all, it is simply a modification of our delta code:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

def payerGamma(notional, swapRate, strike, volatility, maturity,
    discountingRate):
    d1 = (1/(volatility * np.sqrt(maturity)))*(np.log(swapRate/strike) +
        0.5*volatility**2 * maturity)
```

```
    n1 = norm.pdf(d1)
    gamma = notional*np.exp(-discountingRate*maturity)*n1/(swapRate*
       volatility*np.sqrt(maturity))
    return 0.0001*gamma

def plot_gammaVSstrike(notional, swapRate, strike, volatility, maturity,
    discountingRate):
    k = np.linspace(0.8*strike, 1.2*strike, 1000)
    plt.figure(figsize = (12,6))
    plt.plot(k,  payerGamma(notional, swapRate, k, volatility, maturity,
        discountingRate))
    plt.title('Gamma-Strike Relationship')
    plt.xlabel('Strike Rate')
    plt.ylabel('Swaption Gamma')
    plt.show()

plot_gammaVSstrike(1, 0.05, 0.05, 0.02, 1, 0.05)
```

Running the above code yields the following graph:

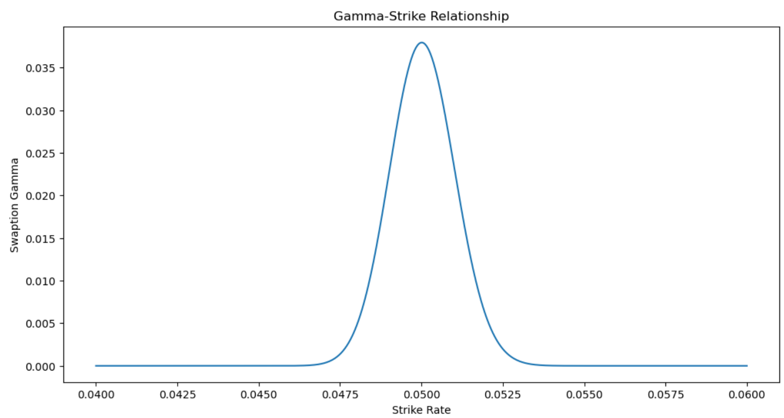

Figure 7.2: Gamma vs Strike Rate for a Payer Swaption

So now we can see that our hypothesis generated from the delta graph was correct. Although it might not seem it due to its second order nature, gamma is one of the most vital greeks, especially when trading volatility through dynamic swaption hedging.

7.3 Vega

7.3.1 Swaption Vega Derivation

Vega is the rate of change of the swaption price with respect to a change in volatility. Mathematically, we are finding:

$$\vartheta = Ne^{-r\tau}(S\partial_\sigma\Phi(d_1) - K\partial_\sigma\Phi(d_2))$$

Let's start by working with $\partial_\sigma\Phi(d_1)$ and then build up from there.

$$\partial_\sigma \Phi(d_1) = \phi(d_1)\partial_\sigma d_1 = \phi(d_1)\partial_\sigma\left(\frac{\ln(\frac{S}{K}) + \sigma^2\tau/2}{\sigma\sqrt{\tau}}\right)$$

Using the quotient rule yields:

$$\partial_\sigma \Phi(d_1) = \phi(d_1)\frac{\sigma\sqrt{\tau}\sigma\tau - (\ln(\frac{S}{K}) + \sigma^2\tau/2)\sqrt{\tau}}{\sigma^2\tau} = \phi(d_1)\frac{\sigma^2\tau - (\ln(\frac{S}{K}) + \sigma^2\tau/2)}{\sigma^2\sqrt{\tau}}$$

Which is then

$$\partial_\sigma \Phi(d_1) = \phi(d_1)\frac{\sigma^2\tau/2 - \ln(\frac{S}{K})}{\sigma^2\sqrt{\tau}}$$

Now we should probably find $\partial_\sigma \Phi(d_2)$ in terms of its d_1-analogue.

$$\partial_\sigma \Phi(d_2) = \phi(d_2)\partial_\sigma d_2 = \phi(d_2)\partial_\sigma(d_1 - \sigma\sqrt{\tau})$$

$$\phi(d_2)\partial_\sigma(d_1 - \sigma\sqrt{\tau}) = \phi(d_2)\partial_\sigma d_1 - \sqrt{\tau}\phi(d_2)$$

So plugging this in,

$$\vartheta = Ne^{-r\tau}(S\phi(d_1)\partial_\sigma d_1 - K\phi(d_2)\partial_\sigma d_1 + K\sqrt{\tau}\phi(d_2))$$

Recall $\phi(d_2) = \frac{S}{K}\phi(d_1)$, we end up with the following formula for vega:

$$\vartheta = Ne^{-r\tau}\phi(d_1)(S\partial_\sigma d_1 - S\partial_\sigma d_2 + \sqrt{\tau}S) = Ne^{-r\tau}S\phi(d_1)\sqrt{\tau}$$

7.3.2 Swaption Vega Python Code

Once again, pricing vega is a relatively simple task in Python and we just have to modify our other functions [1]:

```python
def payerVega(notional, swapRate, strike, volatility, maturity,
    discountingRate):
     d1 = (1/(volatility * np.sqrt(maturity)))*(np.log(swapRate/strike) +
         0.5*volatility**2 * maturity)
     n1 = norm.pdf(d1)
     vega = notional*np.exp(-discountingRate*maturity)*n1*swapRate*np.
        sqrt(maturity)
     return 0.01*vega

def plot_vegaVSstrike(notional, swapRate, strike, volatility, maturity,
    discountingRate):
     k = np.linspace(0.8*strike, 1.2*strike, 1000)
     plt.figure(figsize = (12,6))
     plt.plot(k,  payerVega(notional, swapRate, k, volatility, maturity,
        discountingRate))
     plt.title('Vega-Strike Relationship')
```

[1] I have omitted the imported libraries as they are the same as previous

```
    plt.xlabel('Strike Rate')
    plt.ylabel('Swaption Vega')
    plt.show()

plot_vegaVSstrike(1, 0.05, 0.05, 0.02, 1, 0.05)
```

Running this code will give the following graph:

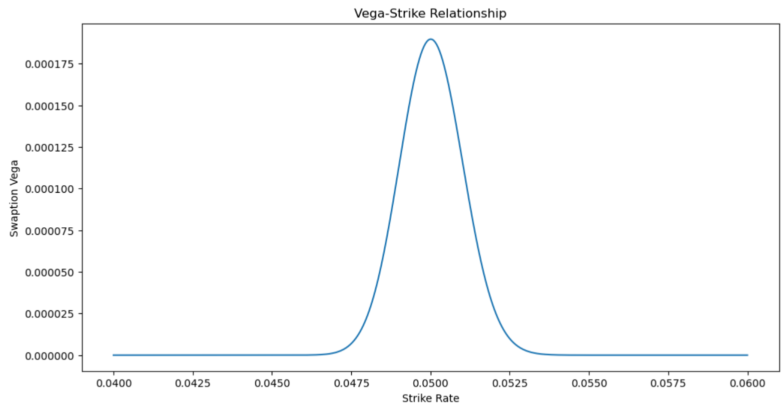

Figure 7.3: Vega vs Strike Rate for a Payer Swaption

7.4 Theta

Theta is a measure of how much my swaption decreases in value as I get closer to maturity. It is:

$$\Theta = N\partial_t e^{-rt}(S\Phi(d_1) - K\Phi(d_2))$$

It is a simple but long derivation so I will leave it as an exercise for you (it is just differentiating each term with respect to time and then adding them up in the correct order), using the previous ideas from earlier derivations.

In short, you should arrive to

$$\Theta = -N\frac{e^{-rT}S\sigma}{2\sqrt{t}}\phi(d_1) + rNe^{-rt}(S\Phi(d_1) - K\Phi(d_2))$$

Here is the Python code for theta:

7.4.1 Swaption Theta Python Code

```
def payerTheta(notional, swapRate, strike, volatility, maturity,
   discountingRate):
    d1 = (1/(volatility * np.sqrt(maturity)))*(np.log(swapRate/strike) +
        0.5*volatility**2 * maturity)
    n1 = norm.pdf(d1)
    N1 = norm.cdf(d1)
```

```
    N2 = norm.cdf(d1 - volatility*np.sqrt(maturity))
    theta = -notional*np.exp(-discountingRate*maturity)*swapRate*
       volatility/(2*np.sqrt(maturity)) + discountingRate*notional*np.
       exp(-discountingRate*maturity)*(swapRate*N1 - strike*N2)
    return theta/(360*maturity)

def plot_thetaVSstrike(notional, swapRate, strike, volatility, maturity,
    discountingRate):
    k = np.linspace(0.8*strike, 1.2*strike, 1000)
    plt.figure(figsize = (12,6))
    plt.plot(k,  payerTheta(notional, swapRate, k, volatility, maturity,
        discountingRate))
    plt.title('Theta-Strike Relationship')
    plt.xlabel('Strike Rate')
    plt.ylabel('Swaption Theta')
    plt.show()

plot_thetaVSstrike(1, 0.05, 0.05, 0.02, 1, 0.05)
```

Plotting this graph shows us that theta is always negative when buying a payer swaption, with time decay increasing in magnitude as we get closer to ATM, and then plateauing at OTM payer swaptions.

8 Swaption Strategies

8.1 Directional Strategies

8.1.1 Outright Payers

Now all of this is good to know, but it is even better to actually be able to use it. Say that I have a high conviction that the Fed have no idea as to how bad rates are and that there will be a severe inflation surprise with the Fed needing to increase rates to 10% in the next 6 months, and then rapidly cutting them back down to 2%. What I can do is buy a payer swaption on FFOIS (the Federal Funds Overnight Index Swap Rate), more specifically the $100mn FFOIS 6mo 2y Payer struck ATM + some spread (say 200 bps). This would cost almost $11,000. If, after 3mo, FFOIS increases to 8.5%, the swaption is worth $2,468,944. However, if the FFOIS remains relatively stable, I will lose all $11,000.

8.1.2 Payer Spread

Now say that I think the Fed will still have to raise rates up, but this time no more than 6.5%. I can buy intro a $100mn 6mo 6mo Payer (ATM, 6.5%). More specifically, this is buying one swaption (the $100mn 6mo 6mo Payer struck ATM) and selling another one (the $100mn 6mo 6mo Payer struck at 6.5%). This strategy is very common in the world of equities and is called a call spread. What a payer spread allows me to do is profit from a move upwards in the underlying rate but only up to a certain level, but as I am capping my maximum profit, I receive premium for selling the payer option.

8.2 Interest Rate Volatility Trading

Recall from vanilla equity options that I can buy a straddle (a call and a put with equivalent maturities), delta hedge through buying/selling the underlying and then trade stock volatility. We can do an equivalent strategy for interest rate volatility through buying swaption straddles.

For example, say that the XYZ rate is at 7% has volatility of 5% per year and that this is an all time volatility high, and I believe that in 6 months, volatility will cool to the long term average of 2%. In order to sell volatility, I sell both a payer and a receiver swaption on the same underlying swap and then delta hedge through buying or selling the underlying swap.

Let us first price these swaptions in Python but first I will state the price and greeks of a receiver swaption:

(i) Price: $Ne^{-r\tau}(K\Phi(-d_2) - S\Phi(-d_1))$

(ii) Delta: $-Ne^{-r\tau}\Phi(-d_1)$

(iii) Gamma: $\frac{Ne^{-r\tau}\phi(d_1)}{S\sigma\sqrt{\tau}}$

(iv) Vega: $Ne^{-r\tau}S\sqrt{\tau}\phi(d_1)$

(v) Theta: $-\frac{e^{-r\tau}S\sigma\phi(d_1)}{2\sqrt{\tau}} + r\text{Price}$

In order to price our straddle, it is as simple as pricing the swaptions separately and then adding them.

```
def swaptionPayerPrice(notional, swapRate, strike,  volatility, maturity
    , discountingRate):
    d1 = (1/(volatility * np.sqrt(maturity)))*(np.log(swapRate/strike) +
        0.5*volatility**2 * maturity)
    N1 = norm.cdf(d1)
    N2 = norm.cdf(d1 - volatility*np.sqrt(maturity))

    price = notional*np.exp(-discountingRate*maturity)*(swapRate*N1 -
       strike*N2)

    return price

def swaptionReceiverPrice(notional, swapRate, strike, volatility,
   maturity, discountingRate):
    d1 = (1/(volatility * np.sqrt(maturity)))*(np.log(swapRate/strike) +
        0.5*volatility**2 * maturity)
    N1 = norm.cdf(-d1)
    N2 = norm.cdf(-d1 + volatility*np.sqrt(maturity))

    price = notional*np.exp(-discountingRate*maturity)*(-swapRate*N1 +
       strike*N2)

    return price

def swaptionStraddlePrice(notional, swapRate, strike, volatility,
   maturity, discountingRate):
    payerPrice = swaptionPayerPrice(notional, swapRate, strike,
       volatility, maturity, discountingRate)
    receiverPrice = swaptionReceiverPrice(notional, swapRate, strike,
       volatility, maturity, discountingRate)
    price = payerPrice + receiverPrice

    return price
```

Running 'swaptionStraddlePrice(1000000, 0.07, 0.07, 0.05, 0.5, 0.05)' tells us the net price of buying:

- \$1mn 6mo 1y ATM Payer Swaption,
- \$1mn 6mo 1y ATM Receiver Swaption

Costing around \$2656. We will be selling these swaptions so we will receive a premium of \$2656. Remember, we aren't trading the price, and that we want to trade volatility and thus we have to delta hedge this position.

Our delta will be the sum of each swaption delta. For the sake of being concise, here is a function that will calculate the price of a swaption straddle and the associated greeks. Running 'swaptionStraddleInfo(1000000, 0.07, 0.07, 0.05, 1, 0.05)' gives an output of:

Swaption Straddle Price: $2656.12

Delta: 18972.31

Gamma: 21678.12

Vega: 531.11

Theta: -8.88

Using the greeks to delta hedge, we can calculate our expected profit from cooling volatility with:

$$\text{Profit} = 10000(\sigma_T - \sigma_0) * \vartheta = 10000(0.05 - 0.02) \times 531 = \$159,300$$

Where we multiply by 10000 to get the number of basis points in the change.

8.3 Modelling Swaption Trades with Python

8.3.1 Methodology

Let us combine what we have learnt in this chapter, together with the 'Synthesis Model'2.3, to study the evolutions of swaptions trades over simulated rate paths. Our model will run as follows:

(i) Simulate sample rate paths under the synthesis model,

(ii) Price different swaption trades under these assumptions,

(iii) Verify that the code has given the correct answer

8.3.2 Modelling Directional Swaption Trades with Python

We first start by stating our swaption conditions. Say we want to study a 1y maturity swaption, with the current swap rate at 5% which is assumed to probably cool to 2% in the long-term. We choose our reversion strength to be 0.11 which is on the extreme upper end of what some papers suggest, as well as volatility being 25% over the year:

```
time = 1
intervals = 500
initialRate = 0.05
longTermRate = 0.02
kappa = 0.11
volatility = 0.25
discountingRate = 0.03
```

```

dt = time/intervals
r = initialRate
rates = []
```

Next, we simulate the swap rate under the synthesis model:

```
for i in range(intervals):
    r0 = r
    r += kappa*(longTermRate - r0)*dt + volatility*np.sqrt(r0)*(np.
       random.standard_normal() - np.random.standard_normal())/100
    rates.append(r)
```

Then with this sample path, we simulate the price of the payer swaption:

```
strike = 0.055
prices = []

for i in range(intervals):
    price = swaptionPayerPrice(1000000, rates[i], strike, volatility,
       time-i*dt, discountingRate)
    prices.append(price)
```

And then finally we plot:

```
plt.figure(figsize = (12,6))
plt.plot(prices)
plt.xlabel(f'Time (1 = {dt} years)')
plt.ylabel('Swaption Price')
plt.show()

plt.figure(figsize = (12,6))
plt.plot(rates, label = 'Simulated Swap Rates')
plt.axhline(strike, color = 'r', label = 'Strike')
plt.xlabel(f'Time (1 = {dt} years)')
plt.ylabel('Interest Rate')
plt.legend()
plt.show()
```

Running this as an example:

We can see that at expiry, our strike is far above the current swap rate and thus the swaptions expire worthless.

I have included the code necessary for the next part in the attached Jupyter Notebook - please let me know if there are any issues with it!

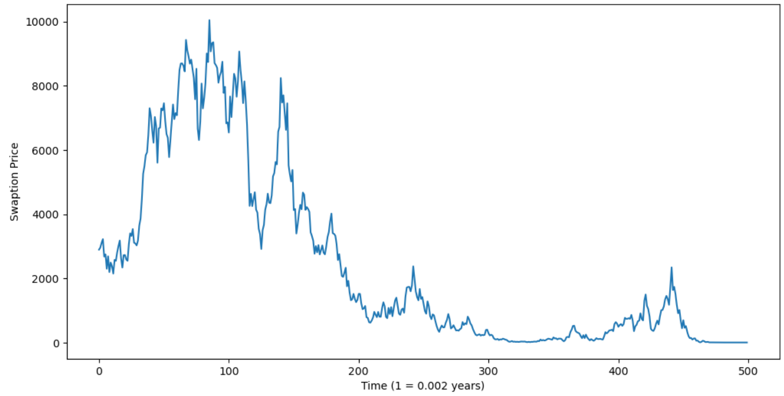

Figure 8.1: Example Payer Swaption Price

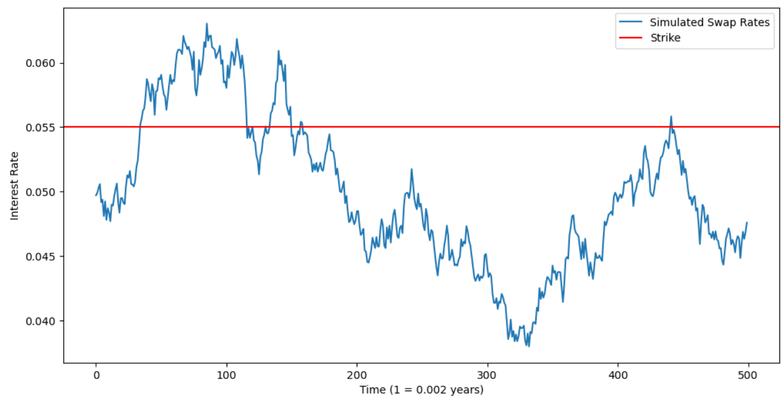

Figure 8.2: Example Swap Rate Path

8.3.3 Modelling Swaption Straddle Prices with Python

Looking at the graphs, we can see that although at no point is the straddle worthless, the initial cost is much higher.

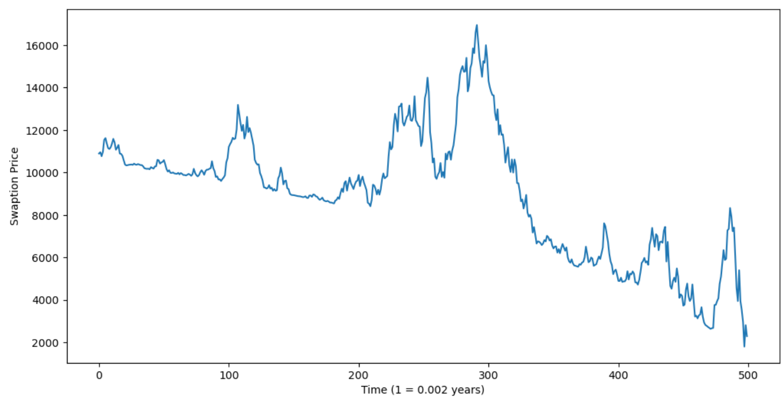

Figure 8.3: Simulated ATM Straddle Prices

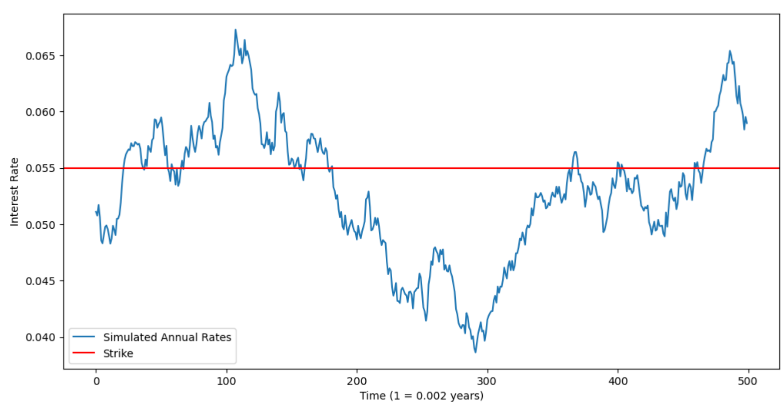

Figure 8.4: Swap Rate Model Output