# A Novel Approach To FX Swap Portfolio Management

## With an Application in Portfolio Optimization

by

Gideon Vissers

to obtain the degree of Master of Science in Applied Mathematics
at the Delft University of Technology,

| | | | |
|---|---|---|---|
| Student number: | 4998081 | | |
| Project duration: | January 8, 2024 – June 25, 2024 | | |
| Thesis committee: | Prof dr. A. Papapantoleon, | TU Delft, | daily supervisor |
| | W. Stijl, | MN | external supervisor |
| | Dr. ir. L. Meester, | TU Delft, | committee member |
| | Dr. F. Fang, | TU Delft, | committee member |

# Preface

In this thesis, we define a new concept of duration for FX Swaps and more broadly for sovereign bonds. The concept of duration already exists for bonds and more specifically coupon bonds, where it is also called ”Macauley Duration”. We aim to define a concept for FX Swaps with similar financial properties and derive mathematical properties from the new definition. A major result of this thesis is the Duration Equivalence Theorem, which states that FX Swap portfolios with the same duration have roughly the same payoff and by extension similar risk exposures. This theorem can be used as a tool for portfolio management and hedging purposes. We also derive a bound of the remainder of the approximation in the theorem so that the applicability of the theorem can be assessed. Based on this first major result, the remainder of the thesis is focused on FX Swap portfolio optimization.

In order to formulate a portfolio optimization problem, we present intuitive and application-oriented models to model a notion of risk and reward within the duration framework. This risk-reward framework is then used to formulate the so-called duration allocation problem based on the Markowitz framework. We then dedicate a chapter to showing that the duration allocation problem can be written as a long-only Markowitz problem and we show a number of properties of this problem and its solutions.

The thesis is concluded by showing how the duration allocation problem can be used in combination with the duration equivalence theorem to optimise FX swap portfolios, which is illustrated by a real-world example. We then validate the novel concepts and methods by simulating a vast number of financial markets, where we compare the resulting optimal portfolio to a number of benchmarks. From these simulations, we can conclude that the duration allocation problem can be used in combination with the duration equivalence to effectively reduce the risk of an FX swap portfolio. The risk reduction of this method compared to the benchmark depends on market conditions, but the results appear promising. We also provide some additional diagnostic tools to assess the risks and rewards of an optimal portfolio.

I would like to thank Antonis Papapantoleon and Wouter Stijl for their support and guidance, Antonis from an academic viewpoint and Wouter from a professional one. I would also like to thank Fang Fang, Tjerk Methorst, Caroline Kortbeek, Gijs Vanoppen, Lander Verlinde and Wasim van Houtum for some insightful brainstorming sessions that helped shape this thesis and Fang Fang and Ludolf Meester for taking the time out of their schedule to be part of the graduation committee and evaluate the thesis. Lastly, I would like to thank the TU Delft and MN for accommodating the writing of this thesis and more specifically MN for granting access to necessary resources.

# Contents

# Chapter 1

# Introduction

In order to diversify financial portfolios, many asset managers decide to invest into both domestic and international markets. In order to invest into foreign assets, an investor needs to first convert their domestic currency into the foreign currency of the country where the market they invest in, is based. For the larger notional volumes asset managers require to convert, these conversions occur through the foreign exchange (also Forex), where $6.6 trillion is traded every day [10]. By converting currencies, the asset manager automatically acquires an exposure to the foreign currency, which carries risks. In order to mitigate these risks, many asset managers use a financial instrument called an "FX Swap" to convert currencies, this instrument is the main subject of this thesis.

## 1.1 FX Swaps

FX swaps are financial products that aim to reduce currency exposures. Normally when converting currencies, an amount of money, for example €1 is converted to the other currency according to the conversion rate, for example $1.08. When the money in the foreign currency has been invested and the investment generates a return, the resulting cash needs to be converted back into the native currency. If in the meantime, the value of the foreign currency has decreased, the return in the native currency could be negative, even if the foreign investment was profitable in the foreign currency. FX swaps aim to remove this currency exposure by fixing the current conversion rate in anticipation of the future conversion back to the native currency. It does so by setting a certain date in the future, called the maturity date. On the maturity date, the currency that is converted today for the current conversion rate (also spot rate), will be converted back for a conversion rate that is decided on today (called the forward rate).

The forward rate is composed of the spot rate and a premium.[1] The premium originates from the time value of money. All currencies change in value over time according to the time value of money, but not at the same rate. In order to account for this difference in time value, a premium needs to be paid out.[2] Because this premium is dependent on the time value of money, FX swap with different maturity dates are priced with a different premium. For this reason, different FX swaps with different maturity dates are traded in the market.

[1] This premium can be positive or negative. Paying a negative premium means receiving the amount of money. Whenever we use the word 'premium', we assume this premium can be negative.

[2] There is another component to the premium called the 'cross currency basis', we will discuss it briefly later, but it will not be a major part of this thesis.

These different FX swap maturities are referred to as the "FX Swap Tenor". Investors can choose to invest in different tenors depending on their expectation of future changes in the market. There are a number of exchange-traded tenors such as 3-month and 6-month tenors, but swap tenors for every day can be traded OTC. The exchange-traded tenors are much more liquid, with the 3-month tenor being the most liquid tenor.

## 1.2 MN: Asset Manager for Pension Funds

MN is an asset manager for a number of dutch pension funds, with a total of €135 billion in assets under management [9]. In order to diversify its portfolios, MN has invested tens of billions of euros into foreign markets such as the US and Japanese markets. Much like many other asset managers, MN uses FX swaps to maintain the vast majority of their foreign currency position, consisting of USD, JPY and GBP. Every day, hundreds of millions of euros worth of FX swap expire and the cash flows from these expiries need to be reinvested. The traders at MN have to choose in which tenors they want to invest each day, depending on their own expectation of market changes, but also based on mandates from the pension funds to which these assets belong. Since MN manages the assets of pension funds, they have more societal responsibilities than many other asset management firms. Because of this, it is critical that MN invests in FX swaps in a risk-responsible manner.

Since FX swap portfolios are meant to mitigate currency exposures and not to generate profit, risk is more central to FX swap portfolio management than returns. When looking at the examples used throughout this thesis, it becomes apparent that the returns to be gained from FX swaps are minimal when compared to the required investment, while the risk of FX swap portfolios can be quite large, especially due to the large volume of swaps often bought. In addition to this intuition for FX swap portfolios in general, due to the societal responsibilities of MN as an asset manager for pension funds, extra care needs to be taken not to invest money irresponsibly and certain mandate requirements need to be met. The mandate requirement that is relevant for this thesis places a restriction on the amount of swaps that are allowed to expire on a single day. Such a restriction is called concentration limit.

## 1.3 Goal: Optimal Allocation

The goal of this thesis is to formulate a method for determining the optimal way to invest in FX swaps each day to minimize risk, taking into account concentration limits and return.

**Problem Statement 1.** *Given a financial market state and an amount of cash, what is the optimal way to allocate cash to FX swap tenors?*

Even though we said that the focus of FX swap investing was on risk and not return, the risk-return payoff should always remain reasonable, so we will take the return component into account. The approach we will take to achieve our goal is to first place the concept of swaps and swap portfolios in a mathematical context, and then study their internal structures. In order to do this, we will formulate a notion for FX swaps called "Duration". The concept of duration is already defined for bonds by Frederick Macaulay in 1938 [12], we will formulate a parallel definition for FX swaps and use the properties of this new concept to formulate a standard portfolio optimization problem for which solution methods are well-studied.

# Chapter 2

# Preliminary Concepts

In this chapter, we introduce some preliminary concepts to motivate the approach we use to place FX trading in a mathematical framework. We mathematically define FX futures and use them to define FX swaps. We then discuss how these FX swaps relate to sovereign bonds. After that, we discuss what kind of interest rates we are interested in and how they work. We conclude with a brief remark regarding the 'cross currency basis'.

## 2.1 FX Trading

FX trading is based around the conversion of currencies. In order to more easily work with this notion of conversion, we create a mathematical framework for FX trading. We do this by adding a unit to the currency value. We use the unit NAT for the native currency and FOR for the foreign currency. This means that if I exchange money today with an exchange rate $C$, the value equality would be

$$X \text{ NAT} = CX \text{ FOR}.$$

In practice, no currency perfectly maintains its value over time. The value of a certain amount of money changes over time. This is called the 'time value of money'. In order to denote this time value of money, we consider the units NAT and FOR to actually correspond to processes $\{\text{NAT}_t\}_{t\geq 0}$ and $\{\text{FOR}_t\}_{t\geq 0}$, where interest rates change the value of the currency over time.

**Definition 1.** ***Interest Rate***
*Any currency with interest rate $r$ can be represented by the process $\{CUR_t\}_{t\geq 0}$, where*

$$CUR_t = CUR_0 e^{rt}.$$

We use log interest rates as opposed to regular interest rates as especially with low interest rates, they are a good approximation for one another. We then just pick the one that has the most mathematical convenience, which is the log return as it imposes additivity of interest rates. We now define the cash portfolio, which is used to track how much of each currency a trader holds at a given time and adjusts the value of the cash in the portfolio for the time value of money.

**Definition 2.** ***Cash Portfolio***
*A cash portfolio $P$ is a function*

$$P(t) = a_t \ NAT_t + b_t \ FOR_t,$$

*where $a_t$ and $b_t$ are real-valued processes.*

## 2.2 FX Futures

One of the most straight-forward derivative on any financial product is the future contract (forward contract for OTC products). FX futures are also traded on the market and have the same intuition as on other assets classes. A future provides the buyer with an obligation to purchase an asset for a predetermined price at some time in the future called the maturity date. We can place this notion into our FX framework.

**Definition 3.** ***FX Future***
*An FX future $F$ with maturity $T$ for a foreign currency $FOR$ at the future conversion rate $C_f$ is given by the payoff*

$$F(T) = -1 \ NAT_0 + C_f \ FOR_0.$$

It can easily be calculated what the future conversion rate should be if the current conversion rate is known.

**Theorem 1.** ***Future Conversion Rate***
*The arbitrage-free future conversion rate of an FX future $F$ with maturity $T$ days where the daily native interest rate is $r_n$, the daily foreign interest rate is $r_f$ and the current conversion rate is $C$ is*

$$C_f = C exp\left(R(r_f - r_n)\right).$$

*Proof.* We can create a replicating portfolio for the FX future by converting $e^{-Tr_n}$ of native currency into foreign currency today. This gives the cash portfolio

$$P(0) = -e^{-Tr_n} \ \mathrm{NAT}_0 + Ce^{-Tr_n} \ \mathrm{FOR}_0.$$

Now at the maturity of the future, we get

$$\begin{aligned} P(T) &= -e^{-Tr_n} \ \mathrm{NAT}_T + Ce^{-Tr_n} \ \mathrm{FOR}_T \\ &= -1 \ \mathrm{NAT}_0 + C\exp\left(T(r_f - r_n)\right) \ \mathrm{FOR}_0. \end{aligned}$$

We see that the native currency payoff of this portfolio is the same as of the future, so the foreign currency payoff should also be the same. Therefore

$$C_f = C\exp\left(T(r_f - r_n)\right)$$

□

This above result is a reformulation of the so-called "Interest Rate Parity" [1]. The FX future plays a crucial role in FX swap pricing, as we will see in the next section.

## 2.3 FX Swaps

FX swaps are financial contracts where one currency is exchanged for another currency today for the present conversion rate and the two currencies are exchanged back at maturity at the future conversion rate.

**Definition 4. *FX Swap***
*An FX swap $F$ with maturity $T$ for a foreign currency $FOR$ with present conversion rate $C_p$ and future conversion rate $C_f$ is given by the initial investment and payoff function respectively*

$$S(0) = -1\ NAT_0 + C_p\ FOR_0$$
$$S(T) = \frac{C_p}{C_f}\ NAT_0 - C_p\ FOR_0.$$

FX swaps can be an effective tool to mitigate currency exposures when investing in foreign markets. The intuition behind this risk coverage can be found in Appendix B. We note here that an FX swap is an example of a swap agreement with only two cash flows, as opposed to certain other swap products that have more cash flows such as dividend swaps or interest rate swaps.

The $S(0)$ component of the FX swap is called the near leg and the $S(T)$ component of the swap is called the far leg. We can decompose the swap into these two components using a future contract.

**Proposition 1. *FX Swap Decomposition***
*An FX swap can be replicated by exchanging to the foreign currency and selling $C_p/C_f$ FX futures with future conversion rate $C_f$.*

The above proposition follows trivially from the definition of FX swaps. Since the FX swap can be replicated in this way, we know that the future conversion rate $C_f$ of the swap is the same as the future conversion rate of an FX future.

**Corollary 1.** *The payoff of an FX swap with present conversion rate $C$ and maturity $T$ is given by*

$$S(T) = exp\left(T(r_n - r_f)\right)\ NAT_0 - C\ FOR_0,$$

*where $r_n$ is the daily native interest rate and $r_f$ is the daily foreign interest rate.*

We see that after the maturity of the FX swap, we no longer have any of the foreign currency. The amount of the native currency we have, has changed however. For every 1 $\mathrm{NAT}_0$ we invest into the swap, we get $\exp\left(T(r_n - r_f)\right)$ back. This means if we invest $X$ $\mathrm{NAT}_0$, letting $\rho = r_n - r_f$, we get $Xe^{\rho T}$ $\mathrm{NAT}_0$ back.

$$X \mapsto Xe^{\rho T}.$$

This portfolio can be replicated by simply purchasing a bond with interest rate $\rho$.

**Definition 5.** ***Net Payoff***
*Consider a financial product with cash flow function $D$, defined on some discrete subset $S$ of $\mathbb{R}$. We define the net payoff of the financial product to be*

$$\pi(D) \coloneqq \sum_{t \in S} D(t).$$

**Definition 6.** ***Bond***
*A bond with maturity $T$ days and daily interest rate $r$ is given by the cash flow function*

$$B(0) = -1$$
$$B(T) = e^{rT}.$$

**Proposition 2.** ***FX Swap as a Bond***
*The net payoff of an FX swap with maturity $T$ and daily native and foreign interest respectively $r_n$ and $r_f$ is the same as the net payoff of a bond with maturity $T$ and daily interest $r = r_n - r_f$.*

This last proposition essentially tells us that we can treat FX swaps as bonds in a mathematical setting. Armed with this knowledge, we discuss which interest rates affect the net payoff of an FX swap in the next section.

## 2.4 Interest Rates

As discussed in the previous section, currencies will change in value according to the interest rate of that currency. The interest rate of a currency is the interest rate provided by the central bank of that currency. In the case of the EUR interest rate, that is the European Central Bank (ECB), for USD it is the Federal Reserve (FED), for CNY it is the People's Bank of China (PBC), etc. Certain financial companies are allowed to place money into an account at a central bank and receive the 'deposit facility' interest rates [4]. In an arbitrage-free framework, the fair interest rates on bonds issued by banks should thus be this deposit facility interest rate.[1]

Central banks hold meetings during which they discuss whether the current interest rates should be changed. In order to make this decision, they consider economic factors such as inflation and employment rates. Changing interest rates also changes the behaviour of market participants, so the central banks aim to change the interest rate to create a stable and healthy economy. In Figure 2.2, we see that both the ECB and the Federal Reserve have increased interest rates at the later period of the graphs. This change was implemented in order to combat the high inflation rate at that time. These interest rate increases had the desired effect and the inflation rate after the increases was lower than before.

In order not to make any general claims about central banks that may not be true for certain specific banks, we focus on the ECB for now. The ECB holds regular meetings, the dates of which are known to the general

[1] In practice, there is arbitrage, so the interest rates may differ slightly from this deposit rate. In practice this difference does not significantly impact the results of this thesis

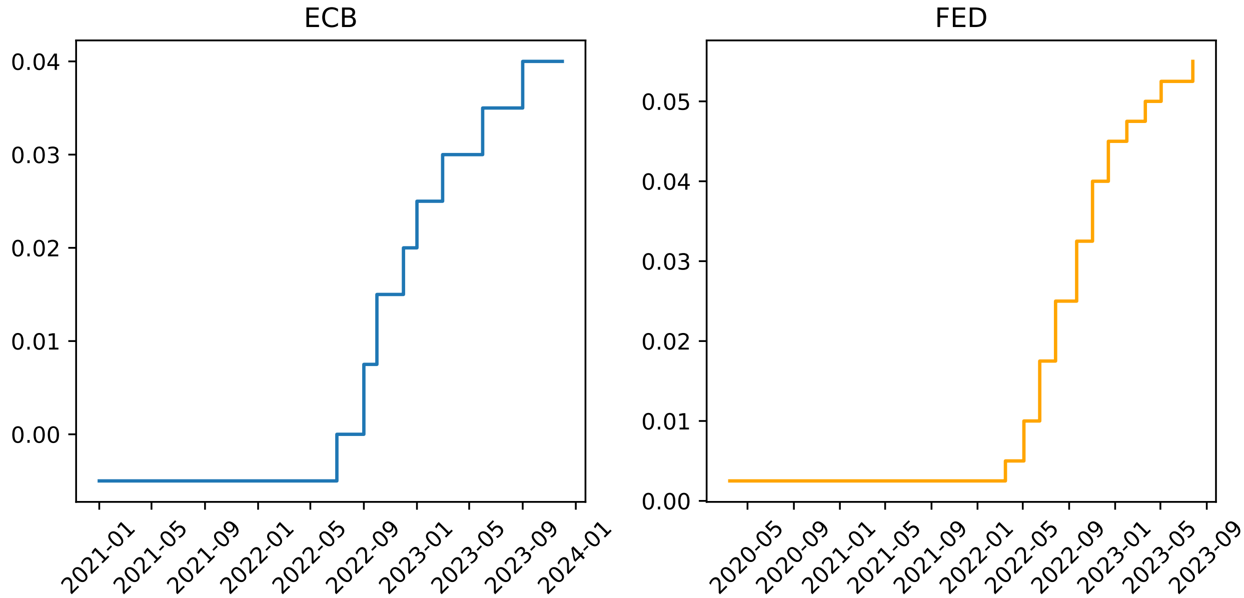


Figure 2.1: Historical EUR and FED Interest Rates.

public beforehand [5]. During a meeting, the ECB decides if they want to increase or decrease the interest rates or leave the interest rates unchanged. In between two meetings, the interest rate does not change.[2] As can be seen in Figure 2.1, these regular meetings create a staircase effect in the historical data. The standard size of an interest rate change for both the ECB and the Federal Reserve is 0.25%, though it is possible for multiple 'jumps' to occur during the same meeting (0.5% or even 0.75% changes). This is not the same for every central bank, as the standard size is 0.1% for the Bank of Japan.

As stated in the previous section, we are not interested in the regular interest rates, but rather the interest rate difference, which can be seen in Figure 2.2. Since the interest rate of either currency does not change between meetings of the corresponding central bank, the difference does not change in between these meetings either.

## 2.5 Cross Currency Basis

In practice the cost of an FX swap is not exactly equal to the interest difference. There is an additional premium that needs to be paid called the 'cross currency basis'. This cross currency basis exists because certain currencies are more desirable than others. Having Mexican Pesos is less desirable than having euros, since the euro is a more universally usable currency. The value of a cross currency basis changes over time and is dependent on many factors. Adding the basis to our models would add significant complexity to the model and make it

[2]During the COVID pandemic, there were some exceptional circumstances during which an inter-meeting interest rate change was implemented, but this we will not consider this possibility.

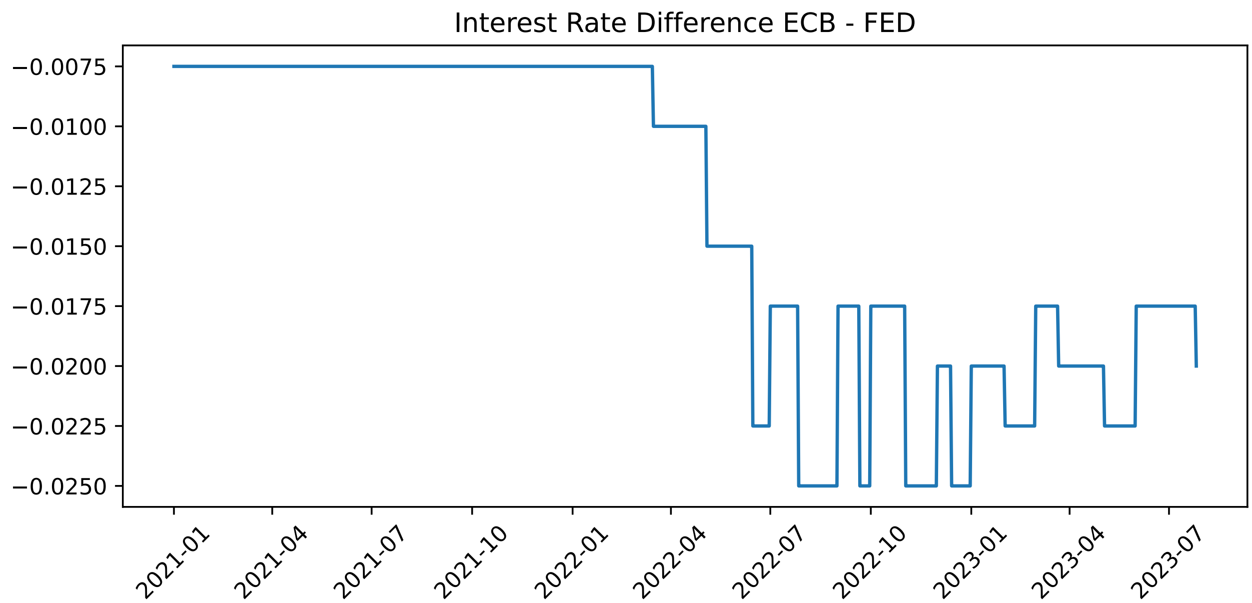


Figure 2.2: Historical Difference Between the ECB and FED Rates.

potentially unusable due to complexity constraints.

In general, changes in the cross currency basis are small compared to the interest rate jumps caused by central banks. There are situations where the cross currency basis could become more volatile, but forecasting such events is well outside the scope of this thesis and so adding such behaviour to our models will not contribute to the quality of our models. Because of these considerations, we completely omit the cross currency basis from our models.

The philosophy behind omitting the cross currency basis is that 'all models are wrong, some are useful' [8], as stated by the statistician George Box. The goal of this thesis is not to produce a model that is 100% accurate, but rather to produce a model that results in useful mathematical and financial results.

# Chapter 3

# Duration

In this chapter, we will discuss the setup of the thesis by defining the concept of duration for sovereign bonds. We will first reiterate the result of the previous chapter stating that FX swaps can be modelled as bonds. We then provide some definitions and intuitions surrounding sovereign bonds. We proceed to build a bridge between the pre-existing concept of duration and the new definition we will give, upon which we will formally define the concept of duration for sovereign bonds and FX swaps. We conclude by proving a number of properties regarding duration and related concepts.

## 3.1 FX Swaps as Bonds

In the previous chapter, we proved that the payoff of an FX swap is the same as the payoff of a bond. This means we can model FX swaps as bonds where the interest rate of the bond is the difference of the interest rates between the two currencies. These currency interest rates coincide with the interest rates set by central banks and are thus equal to the interest rates of sovereign bonds. For this reason, we model FX swaps as sovereign bonds.

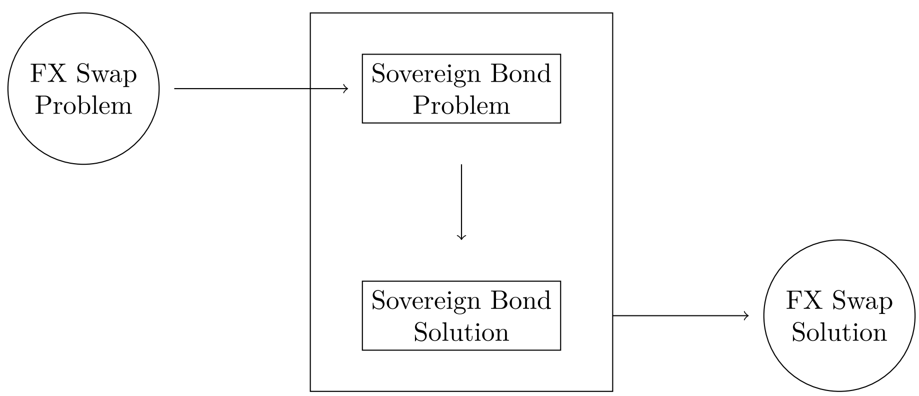


Figure 3.1: Diagram of Solving FX Swap Problems

We note here that FX swaps and bond are **not** the same financial product. They merely have the same payoff structure and can thus be modelled the same way **in a mathematical context**. In a real-world context, bonds and swaps fulfill two completely different objectives and have completely different financial implications. The approach we take to solve FX swap-related problems is visualised in Figure D.1. We see here that when faced with an FX swap problem, we can first formulate the problem mathematically using the mathematical framework for FX swaps. We then translate this mathematical FX swap problem into a mathematical sovereign bond problem. We then solve the sovereign bond problem mathematically and then translate the solution for sovereign bonds into a solution that can be directly applied to FX Swaps. This approach can be compared to other mathematical problem solving methods such as transforming a temporal problem into a frequency problem to apply Fourier analysis or solving real-valued problem using inferences obtained in the complex domain, which is a common method for analysing certain differential equation problems.

## 3.2 Bonds

In the previous chapter, we defined what a bond is in terms of cash flows. We now redefine the concept of bonds using the value of the bond at any given time as an anchor point. We note here that we use the time scale of days. This means the unit of any time measurement is days and any interest rate corresponds to a daily interest rate.

**Definition 7.** ***Bond***
*A bond $B$ with interest rate $r$, purchase date $a \in \mathbb{N}$ and maturity $b \in \mathbb{N}$ is a function $B : \mathbb{N} \to \mathbb{R}$ such that for $t \in \mathbb{N}$*[a]

$$B(t; r, (a, b]) = \begin{cases} 1, & t \leq a \\ e^{r(t-a)}, & a < t \leq b \\ e^{r(b-a)}, & b < t. \end{cases}$$

*We call $(a, b]$ the running time of the bond.*

[a]Note that by choosing $t \in \mathbb{N}$, we are committing to a discrete framework. This choice of framework is to keep the derivations as simple and intuitive as possible.

We call $B(T; r, (a, b])$ the payoff of the bond for any $T \geq b$. We note here that we only consider the value of the bond at discrete points in time. This is simply a choice of convenience as this simplifies the mathematics. Also note that if $a > 0$, purchasing that bond corresponds to purchasing a future for a bond with maturity $b - a$. The notation for bonds provided in the definition above is rather messy. To simplify the notation, we follow some notation convention.

**Convention 1.** *If the interest rate is either known from the context or not relevant and if the time-dependence is not relevant, then we abbreviate*

$$\mathcal{B}(a, b) \coloneqq B(t; r, (a, b])$$
$$\mathcal{B}_i \coloneqq B(t; r, (0, i])$$

Note that when we want to make claims about a bond of the form $\mathcal{B}(a, b)$, often times, we can shift the time axis to make claims about $\mathcal{B}_{b-a}$.

We now want to deconstruct a bond into smaller components. We do this by using overnight bonds.

**Definition 8.** *An overnight bond with interest rate $r$ and purchase date $\tau$ is the bond $b(t; r, \tau) := B(t; r, (\tau, \tau + 1])$.*

A longer running bond can now be constructed by purchasing consecutive overnight bonds.

**Proposition 3.** *A bond with running time $(a, c]$ and interest rate $r$ can be written as the product of $c - a$ overnight bonds with interest rate $r$.*

$$\mathcal{B}(a, c) = \prod_{i=a+1}^{c} b(t; r, i).$$

*Proof.* The proof follows from the definitions. For $t \leq a$, we have

$$\mathcal{B}(a, c) = 1 \qquad\qquad b(t; r, i)|i \geq a = 1.$$

For $a < t \leq c$, we have

$$\begin{aligned} \mathcal{B}(a, c) &= e^{r(t-a)} \\ b(t; r, i)|i \leq t &= e^{r} \\ b(t; r, i)|i > t &= 1, \end{aligned}$$

so clearly the formula holds. For $c \leq t$, we have

$$\begin{aligned} \mathcal{B}(a, c) &= e^{r(t-a)} \\ b(t; r, i)|i < c &= e^{r}, \end{aligned}$$

and again we see that the formula holds. □

The above proposition provides us with a formula for a longer-running bond using overnight bonds with the same interest rates. In practice, since these overnight bonds have different running times, they are likely to have different interest rates. The next proposition provides us with a method for constructing a longer-running bond using overnight bonds with different interest rates.

**Proposition 4.** ***Bond Decomposition***
*A payoff of a bond with running time $[a, c)$ and interest rate $\rho$ can be written as the product of the payoffs of $c - a$ overnight bonds with interest rate process $r : \mathbb{N} \to \mathbb{R}$.*

$$B(c; \rho, (a, c]) = \prod_{i=a+1}^{c} b(c; r(i), i),$$

*where*[a]

$$\rho = \frac{\sum_{i=a+1}^{c} r(i)}{c - a}.$$

[a]Note that in a continuous context, it is customary to set the effective interest rate to be $\int_0^T r_t \, dt$, where $r_t$ is a continuous interest rate process. The given formula is simply the average of the discretised counterpart.

*Proof.* The payoff of the bond is

$$\begin{aligned}
\mathcal{B}(a, b) &= e^{\rho(c-a)} \\
&= \exp\left(\sum_{i=a+1}^{c} r(i)\right) \\
&= \prod_{i=a+1}^{c} \exp\left(r(i)\right) \\
&= \prod_{i=a+1}^{c} b(c; r(i), i).
\end{aligned}$$

$\square$

If we now call the effective interest rate of a bond $\mathcal{B}_T$ the cumulative interest rate over the entire period of the bonds, so

$$r_e = rT,$$

then we see that the effective interest of a bond $\mathcal{B}_T$ in the Bond Decomposition proposition above is given by

$$\rho_e = \sum_{i=1}^{T} r(i).$$

Now if the interest rate process $r(i)$ is known for every $i$, then we can simply plug in the formula to determine the fair $\rho_e$. In practice however, interest rate changes can not always be predicted with certainty. Despite this, longer-running bonds are still traded with a certain fixed interest rate that is fixed at time $t = 0$. This traded interest rate is determined by what the market predicts the interest rate is going to do. If we define $\mathbb{E}_m$ to be the market expectation, we can write the effective interest rate as

$$\rho_e = \sum_{i=1}^{T} \mathbb{E}_m(r(i)).$$

We call the interest rate of the above form 'conform with market expectations'. Now instead of purchasing this one long-running bond, we can also simply purchase the overnight bond every day, in which case the effective payoff would be

$$\rho_e = \sum_{i=1}^{T} r(i).$$

We see that the first effective interest is deterministic and the second effective interest is stochastic.

## 3.3 Interest Adjusted Bond Payoff

In order to compare the payoffs of longer-running bonds with shorter-running bonds, we first need to regularise their payoff. After all, when a shorter-running bond expires, new bonds can be bought to increase the overall payoff.

**Definition 9.** ***payoff***
*The Let $\mathcal{B}(a,b)$ be a bond with interest rate $\rho$ in a market with interest rate process $r : \mathbb{N} \to \mathbb{R}$ over the span of $T$ days. The interest-adjusted payoff (IAP) of the bond is defined as*

$$IAP(\mathcal{B}(a,b)) := exp\left(\sum_{i=1}^{a} r(i) + \rho(b-a) + \sum_{i=b+1}^{T} r(i)\right).$$

This definition essentially states that outside the running time of the bond, we purchase overnight bonds to increase the overall payoff. Since we assume a simplified interest model when dealing with sovereign bonds, we can further simplify this payoff formula by combining this simplified formula with last section's intuition that the bond price coincides with the market expectation.

**Proposition 5.** ***Sovereign Bond Payoff***
*Let $\mathcal{B}(0,\tau)$ be a in a market with interest rate process $r : \mathbb{N} \to \mathbb{R}$ over the span of $T$ days and let the interest rate of $\mathcal{B}$ be conform with market expectations. Let $I_1, I_2, \ldots, I_N$ be a sequence of intervals that partition $(0,T]$. If the interest rate process is constant on each $I_k$, say $r(t) = r_k$ for $t \in I_k$, then*

$$IAP(\mathcal{B}(0,\tau)) = exp\left(\sum_{k=1} |(0,\tau] \cap I_k| \mathbb{E}\left[r_k\right] + |(0,\tau]^C \cap I_k| r_k\right).$$

*Proof.* We assume last section's intuition that $\rho = \sum_{i=1}^{\tau} \mathbb{E}\left[r(i)\right]/\tau$, we then get

$$\begin{aligned} IAP(\mathcal{B}(0,\tau)) &= \exp\left(\rho\tau + \sum_{i=\tau+1}^{T} r(i)\right) \\ &= \exp\left(\sum_{i=1}^{\tau} \mathbb{E}\left[r(i)\right] + \sum_{i=1}^{T} r(i)\right) \end{aligned}$$

We can now group the period with constant interest rates together.

$$= \exp\left(\sum_{k=1}^{N}\sum_{i\in I\cap(0,\tau]}\mathbb{E}\left[r_k\right]+\sum_{k=1}^{N}\sum_{i\in I\setminus(0,\tau]} r_k\right)$$
$$= \exp\left(\sum_{k=1}|(0,\tau]\cap I_k|\mathbb{E}\left[r_k\right]+|(0,\tau]^C\cap I_k|r_k\right).$$

□

The intuition in the above formula is that by purchasing a bond, we are 'fixing' the first $\tau$ interest rate units as the expected interest rate.

## 3.4 Motivation and Duration in Coupon Bonds

The concept of duration was first defined by Frederick Macaulay is 1938 [12]. Intuitively speaking, the concept of duration aims to measure the weighted average time to maturity for the cash flows of a bond. Duration can be used to manage larger bond portfolios in a more intuitive and simplified way. The additional benefit of duration is that it provides a measure for the sensitivity of the cash flows of a bond portfolio to changes in the interest rate.

The currently used mathematical definition of Macaulay's duration of a bond $\mathcal{B}$ is [7]

$$\text{Dur}\left(\mathcal{B}\right)\times \text{PV}(\mathcal{B}) = \sum_{n=1}^{N}\frac{t\times C}{(1+y)^t}+\frac{N\times M}{(1+y)^N}$$

In this formula, $\text{PV}(\mathcal{B})$ represents the present value of the bond, $C$ is the periodic coupon payout, $N$ is the amount of periods, $y$ is the periodic yield, $t$ is the respective time period and $M$ is the maturity value. We note that FX swaps do not have coupon payments, so the bonds we use to model swaps also do not. For this reason, $C=0$ and $N=1$, so

$$\text{Dur}\left(\mathcal{B}\right)\times \text{PV}(\mathcal{B}) = \frac{M}{1+y}$$

Now for a zero-coupon bond, the present value of the bond is simply equal to $M/(1+y)$, so the duration is equal to 1. Clearly this definition of duration is not very useful for zero-coupon bonds, so we look for an alternative definition that has the same nice properties as the existing notion of duration. Remember that one of the intuitions of duration is that it represents the sensitivity of the bond value to the interest rate. Note that if we purchase a bond today, the interest adjusted payoff is given by Proposition 5 as

$$IAP(\mathcal{B}(0,\tau)) = \exp\left(\sum_{k=1}|(0,\tau]\cap I_k|\mathbb{E}\left[r_i\right]+|(0,\tau]^C\cap I_k|r_i\right).$$

We now want to calculate the sensitivity of the bond payout to the interest rate, but we note that each period has its own interest rate. We also note that the interest rates $r_i$ only become known at the central bank meetings and so their value does not change day-to-day, making sensitivity calculations to this rate not very useful. The expected interest rate $\mathbb{E}[r_i]$ can change on a daily basis however, so we calculate the sensitivity of the bond payout to this rate and can do so for each period.

$$\frac{dIAP(\mathcal{B})}{d\mathbb{E}[r_i]} = |(0,\tau] \cap I_k| IAP(\mathcal{B}).$$

Just like Macaulay, we now normalize this sensitivity using the current expected payout of the bond, so the normalised sensitivity is $|(0,\tau] \cap I_k|$. We will call this quantity the duration of $\mathcal{B}$ in the period $I_k$.

We now note first of all that this is a completely new definition and as such we will need to analyse its properties from scratch. We secondly note that this thesis only discusses this definition of duration in the context of FX swaps and zero-coupon sovereign bonds. This definition does not clash with Macauley's definition as both concepts are defined for distinctly separate financial products and neither is compatible with the financial product that the other is defined for.

## 3.5 Duration and Termination

Now that we have explained the intuition behind our new concept of duration, we create a completely rigorous framework to work in. Creating this framework will allow us to prove theorems with more rigour, ensuring an airtight theory. We first tackle the ambiguity of the interpretation of time in our framework.

**Definition 10.** ***Discrete Interval***
*A discrete interval between $a$ and $b$ where $a < b$ is defined by*

$$\overline{(a,b]} := (a,b] \cap \mathbb{Z}.$$

*We define the length of this discrete interval as $|\overline{(a,b]}| = b - a$.*

For concepts like interest-adjusted payoff, we restrict ourselves to a time horizon up to some time $T \in \mathbb{N}$. We now define this time horizon to be the discrete interval $\mathbb{T} = \overline{(0,T]}$. Whenever we refer to a period $I$, we mean a discrete subinterval $I = \overline{(a,b]} \subseteq \mathbb{T}$. We now define the duration of a bond.

**Definition 11.** ***Duration***
*The duration of a bond $\mathcal{B}(a,b)$ in a period $I = \overline{(t_1,t_2]}$ is the length of the intersection*

$$Dur(\mathcal{B}(a,b), I) := |\overline{(a,b]} \cap \overline{(t_1,t_2]}|.$$

We illustrate the above definition with an example. Imagine the time horizon is $\mathbb{T} = \overline{(0,60]}$ with $I_1 = \overline{(0,30]}, I_2 = \overline{(30,60]}$. A 50-day bond then has the durations

$$\begin{aligned} \text{Dur}(\mathcal{B}_{50}, I_1) &= 30 \\ \text{Dur}(\mathcal{B}_{50}, I_2) &= 20. \end{aligned}$$

In practice, we are often interested in looking at portfolios of bonds as opposed to just a single bond. For our use case, we assume that we can only purchase bonds and not sell them, in which case we will refer to bond portfolios as bond investments.

**Definition 12.** ***Bond Investment***
*A bond investment over the time horizon $\mathbb{T}$ with capital limit $C$ is a vector $\xi \in \mathbb{R}^T$ such that*

$$\sum_{i=1}^{T} \xi_i = C.$$

We now simply define the duration of a bond investment to be equal to the duration of each bonds in the investment.

**Definition 13.** ***Duration of a Bond Investment***
*The duration of a bond investment $\xi$ in a period $I \subseteq \mathbb{T}$ is the sum of the durations of the bonds in the bond investment*

$$Dur(\xi, I) = \sum_{i=1}^{T} \xi_i \ Dur(\mathcal{B}_i, I)$$

We now have a way to study bonds separately using duration and a way to study bonds collectively using bond investments. Ideally, we would also like have a way to study bonds within a certain period. To do this, we define the restricted bond investment.

**Definition 14.** ***Restricted Bond Investment***
*Let $\xi$ be a bond investment with time horizon $\mathbb{T}$ and let $I$ be a period in $\mathbb{T}$. We define $\xi$ restricted to $I$ as the vector $(\xi|I)$ with components*

$$(\xi|I)_i = \begin{cases} 0, & i \notin I \\ \xi_i, & i \in I. \end{cases}$$

Since we are interested in studying the bonds that expire in a given period, it is also interesting to define a function for determining if and how many bonds expire in a period. We call such a function the 'termination function'.

**Definition 15.** ***Termination***
*We define the termination of a bond $\mathcal{B}(a, b)$ within a period $I$ as*

$$Term(\mathcal{B}(a, b), I) = \mathbb{1}(b \in I).$$

*We define the termination of a bond investment $\xi$ in a period $I$ as*

$$Term(\xi, I) = \sum_{i=1}^{T} \xi_i \ Term(\mathcal{B}_i, I)$$

Now that we have defined the basic functions that act as the foundation of this thesis, the next section is dedicated to deriving some basic properties of these functions.

## 3.6 Basic Properties of Duration and Termination

The first property we show is a reduction of the termination formula for bond investments.

**Proposition 6.** ***Termination of a bond Investment***
*The termination of a bond investment $\xi$ within a period $I$ is given by*

$$\mathit{Term}\,(\xi, I) = \sum_{i \in I} \xi_i.$$

*Proof.* Let $\xi$ be a bond investment, then

$$\begin{aligned}\mathrm{Term}\,(\xi, I) &= \sum_{i=1}^{T} \xi_i \; \mathrm{Term}\,(\mathcal{B}_i, I) \\ &= \sum_{i=1}^{T} \xi_i \; \mathbb{1}(i \in I) \\ &= \sum_{i \in I} \xi_i.\end{aligned}$$

□

It is easy to see that this also means that

$$\mathrm{Term}\,(\xi, I) = \sum_{i=1}^{T} (\xi|I)_i$$

and

$$\begin{aligned}\mathrm{Term}\,((\xi|I), I) &= \mathrm{Term}\,(\xi, I) \\ I \cap J = \emptyset \implies \mathrm{Term}\,((\xi|I), J) &= 0.\end{aligned}$$

We now prove some additivity properties of duration and termination.

**Proposition 7.** ***Additivity of Duration***
*Let $\xi$ and $\zeta$ be two bond investments and let $I \cap J = \emptyset$, then*

$$\begin{aligned}\mathit{Dur}\,(\xi + \zeta, I) &= \mathit{Dur}\,(\xi, I) + \mathit{Dur}\,(\zeta, I) \\ \mathit{Dur}\,(\xi, I \cup J) &= \mathit{Dur}\,(\xi, I) + \mathit{Dur}\,(\xi, J)\,.\end{aligned}$$

*Proof.* We first prove the first additivity property. Let $\xi$ and $\zeta$ be two bond investments. Then

$$\begin{aligned}\mathrm{Dur}\,(\xi + \zeta, I) &= \sum_{i=1}^{T} (\xi_i + \zeta_i) \mathrm{Dur}\,(\mathcal{B}_i, I) \\ &= \sum_{i-1}^{T} \xi_i \; \mathrm{Dur}\,(\mathcal{B}_i, I) + \sum_{i-1}^{T} \zeta_i \; \mathrm{Dur}\,(\mathcal{B}_i, I) \\ &= \mathrm{Dur}\,(\xi, I) + \mathrm{Dur}\,(\zeta, I)\end{aligned}$$

Now for the second additivity property, let $I \cap J = \emptyset$. Then

$$
\begin{aligned}
\mathrm{Dur}\left(\xi, I \cup J\right) &= \sum_{i=1}^{T} \xi_i \ \mathrm{Dur}\left(\mathcal{B}_i, I \cup J\right) \\
&= \sum_{i=1}^{T} \xi_i \ |\overline{(0,i]} \cap (I \cup J)| \\
&= \sum_{i=1}^{T} \xi_i \left( |\overline{(0,i]} \cap I| + |\overline{(0,i]} \cap J)| \right) \\
&= \sum_{i=1}^{T} \xi_i \ \mathrm{Dur}\left(\mathcal{B}_i, I\right) + \sum_{i=1}^{T} \xi_i \ \mathrm{Dur}\left(\mathcal{B}_i, J\right) \\
&= \mathrm{Dur}\left(\xi, I\right) + \mathrm{Dur}\left(\xi, J\right).
\end{aligned}
$$

□

**Proposition 8.** ***Additivity of Termination***
*Let $\xi$ and $\zeta$ be two bond investments and let $I \cap J = \emptyset$, then*

$$
\begin{aligned}
\mathit{Term}\left(\xi + \zeta, I\right) &= \mathit{Term}\left(\xi, I\right) + \mathit{Term}\left(\zeta, I\right) \\
\mathit{Term}\left(\xi, I \cup J\right) &= \mathit{Term}\left(\xi, I\right) + \mathit{Term}\left(\xi, J\right).
\end{aligned}
$$

*Proof.* We first prove the first additivity property. Let $\xi$ and $\zeta$ be two bond investments. Then

$$
\begin{aligned}
\mathrm{Term}\left(\xi + \zeta, I\right) &= \sum_{i \in I} (\xi_i + \zeta_i) \\
&= \sum_{i \in I} \xi_i + \sum_{i \in I} \zeta_i \\
&= \mathrm{Term}\left(\xi, I\right) + \mathrm{Term}\left(\zeta, I\right)
\end{aligned}
$$

For the second property, let $I$ and $J$ be two disjoint periods. Then

$$
\begin{aligned}
\mathrm{Term}\left(\xi, I \cup J\right) &= \sum_{i \in I \cup J} \xi_i \\
&= \sum_{i \in I} \xi_i + \sum_{i \in J} \xi_i \\
&= \mathrm{Term}\left(\xi, I\right) + \mathrm{Term}\left(\xi, J\right).
\end{aligned}
$$

□

**Proposition 9.** ***Scaling***
*Let $\xi$ be a bond investment and let $s > 0$, then*

$$\begin{aligned} Dur\,(s \cdot \xi, I) &= s \cdot Dur\,(\xi, I) \\ Term\,(s \cdot \xi, I) &= s \cdot Term\,(\xi, I) \end{aligned}$$

*Proof.* We prove both statements separately.

$$\begin{aligned} \mathrm{Dur}\,(s \cdot \xi, I) &= \sum_{i=1}^{T} s\xi_i \mathrm{Dur}\,(\mathcal{B}_i, I) \\ &= s \sum_{i=1}^{T} \xi_i \mathrm{Dur}\,(\mathcal{B}_i, I) \\ &= s \cdot \mathrm{Dur}\,(\xi, I) \end{aligned}$$

For the termination, we have

$$\begin{aligned} \mathrm{Term}\,(s \cdot \xi, I) &= \sum_{i \in I} s\xi_i \\ &= s \sum_{i \in I} \xi_i \\ &= s \cdot \mathrm{Term}\,(\xi, I)\,. \end{aligned}$$

□

## 3.7 Effective Duration and Duration Complement

As previously said, we are interested in the collection of bonds that expire in a single period. In order to study the duration of those bonds in the given period, we define the concept of effective duration.

**Definition 16.** ***Effective Duration***
*The effective duration of a bond $\mathcal{B}(a, b)$ in a period $I$ is equal to the duration of that bond if the bond expires in that period, else it is zero*

$$eDur\,(\mathcal{B}(a, b), I) = Dur\,(\mathcal{B}(a, b), I)\ Term\,(\mathcal{B}(a, b), I)$$

*The effective duration of a bond investment $\xi$ in a period $I$ is the sum of the effective durations of the bonds in the bond investment*

$$eDur\,(\xi, I) = \sum_{i=1}^{T} \xi_i\ eDur\,(\mathcal{B}_i, I)$$

There are a number of properties that relate different time periods to other, disjoint periods by way of duration. In order to formalise these properties, we define a partition of the time horizon $\mathbb{T}$, to ensure pairwise disjointedness of periods.

**Definition 17.** ***Time Partition***
*Let $\mathbb{T}$ be a time horizon. A time partition $\mathcal{I} = \{I_i\}$ of $\mathbb{T}$ is a partition of $\mathbb{T}$ consisting of discrete intervals.*

We say that $\xi$ is a bond investment over a time partition $\mathcal{I}$ if $\xi$ is a bond investment over a time horizon $\mathbb{T}$ and $\mathcal{I}$ is a time partition of $\mathbb{T}$. In section 2.4, we stated that central banks like the ECB hold regular meetings in between which interest rates do not change. If we define the time partition to consist of the periods between meetings, we can say that interest rates remain constant on each period in the partition.

**Definition 18.** ***Partitioning Process***
*Let $\mathbb{T}$ be a time horizon with time partition $\mathcal{I}$ and let $\{r_t\}$ be a stochastic process on $\mathbb{T}$. If $\{r_t\}$ is constant on each period $I_k$, then we define the partitioning process as the function $R : \mathcal{I} \to \mathbb{R}$ so that*

$$R_k = r_t, t \in I_k.$$

We now provide a number of properties connecting duration, termination and effective duration.

**Proposition 10.** ***Duration Properties***
*Let $\xi$ be a bond investment over the time horizon $\mathbb{T}$, let $\mathcal{I} = \{I_i\}$ be a time partition of $\mathbb{T}$ and let $\tau \in \mathbb{T}$.*

1. *If $i < j$ and $Term(\mathcal{B}_\tau, I_j) = 1$, then $Dur(\mathcal{B}_\tau, I_i) = |I_i|$.*
2. *If $i < j$ and $Term(\mathcal{B}_\tau, I_i) = 1$, then $Dur(\mathcal{B}_\tau, I_j) = 0$.*
3. *If $I_i = \overline{(a_i, b_i]}$, then effective duration can be simplified to*

$$\begin{aligned} eDur(\xi, I_i) &= \sum_{k \in I_i} \xi_k \; Dur(\mathcal{B}_k, I_i) \\ &= \sum_{k \in I_i} \xi_k \; (k - a_i) \end{aligned}$$

4. *If $I_i = \overline{(a_i, b_i]}$, then*

$$Dur(\xi, I_i) = \sum_{k=a_i+1}^{T} \xi_k \; Dur(\mathcal{B}_k, I_i)$$

5. *For any $i$,*

$$Dur(\xi, I_i) = eDur(\xi, I_i) + |I_i| \, Term\left(\xi, \bigcup_{k=i+1}^{N} I_k\right)$$

*Proof.* We prove each statement separately.

1. If $i < j$ and Term $(\mathcal{B}_\tau, I_j) = 1$, then the bond running period is $(0, \tau]$, where $\tau \in I_j = \overline{(a_j, b_j]}$. Since $i < j$,

we have that $b_i \le a_j < \tau$, so $I_i \subseteq \overline{(0,\tau]}$, therefore

$$\begin{aligned}\mathrm{Dur}\left(\mathcal{B}_\tau, I_i\right) &= \left|\overline{(0,\tau]} \cap I_i\right| \\ &= |I_i|\end{aligned}$$

2. If $i < j$ and $\mathrm{Term}\left(\mathcal{B}_\tau, I_i\right) = 1$, then the bond running period is $[0,\tau)$ with $\tau \in I_i = \overline{(a_i, b_i]}$. Since $i < j$, we have that $\tau \le b_i \le a_j$, so $\overline{(0,\tau]} \cap I_j = \emptyset$.

3. We use the definition

$$\begin{aligned}\mathrm{eDur}\left(\xi, I_i\right) &= \sum_{k=1}^{T} \xi_k \ \mathrm{eDur}\left(\mathcal{B}_k, I_i\right) \\ &= \sum_{k=1}^{T} \xi_k \ \mathrm{Dur}\left(\mathcal{B}_k, I_i\right) \mathrm{Term}\left(\mathcal{B}_k, I_i\right) \\ &= \sum_{k=1}^{T} \xi_k \ \mathrm{Dur}\left(\mathcal{B}_k, I_i\right) \mathbb{1}(k \in I_i) \\ &= \sum_{k \in I_i} \xi_k \ \mathrm{Dur}\left(\mathcal{B}_k, I_i\right) \\ &= \sum_{k \in I_i} \xi_k \ \left|\overline{[0,i)} \cap \overline{(a_i, b_i]}\right| \\ &= \sum_{k \in I_i} \xi_k \ (i - a_i).\end{aligned}$$

4. We use the definition

$$\begin{aligned}\mathrm{Dur}\left(\xi, I_i\right) &= \sum_{k=1}^{T} \xi_k \ \mathrm{Dur}\left(\mathcal{B}_k, I_i\right) \\ &= \sum_{k=1}^{a_i} \xi_k \ \mathrm{Dur}\left(\mathcal{B}_k, I_i\right) + \sum_{k=a_i+1}^{T} \xi_k \ \mathrm{Dur}\left(\mathcal{B}_k, I_i\right)\end{aligned}$$

Now $\forall k < a_i : \exists j < i : k \in I_j$. This means that for this $k$, $\mathrm{Term}\left(\mathcal{B}_k, I_j\right) = 1$, so by (2): $\mathrm{Dur}\left(\mathcal{B}_k, I_i\right) = 0$, therefore the first sum is equal to 0.

5. We write out the duration

$$
\begin{aligned}
\mathrm{Dur}\left(\xi, I_i\right) &= \sum_{k=1}^{T} \xi_k \ \mathrm{Dur}\left(\mathcal{B}_k, I_i\right) \\
(4) &= \sum_{k=a_i+1}^{b_i} \xi_k \ \mathrm{Dur}\left(\mathcal{B}_k, I_i\right) + \sum_{k=b_i+1}^{T} \xi_k \ \mathrm{Dur}\left(\mathcal{B}_k, I_i\right) \\
(1,3) &= \mathrm{eDur}\left(\xi, I_i\right) + \sum_{k=b_i}^{T} \xi \ |I_i| \\
(\text{additivity}) &= \mathrm{eDur}\left(\xi, I_i\right) + \mathrm{Term}\left(\xi, \bigcup_{k=i+1}^{N} I_k\right)
\end{aligned}
$$

□

We now also provide a number of properties for restricted bond investments.

**Proposition 11. *Restricted Bond Investment Properties***
*Let $\mathbb{T}$ be a time horizon with time partition $\mathcal{I} = \{I_i\}$ and let $\xi$ be a bond investment over $\mathbb{T}$. The following properties hold*

1. *If $s \neq k$, then*

$$
eDur\left((\xi, I_k), I_s\right) = 0.
$$

2. *If $s < k$, then*

$$
Dur\left((\xi|I_k), I_s\right) = |I_s| \sum_{i \in I_k} \xi_i.
$$

3. *For any $k$,*

$$
Dur\left((\xi|I_k), I_k\right) = eDur\left(\xi, I_k\right)
$$

4. *If $k < s$, then*

$$
Dur\left((\xi|I_k), I_s\right) = 0
$$

*Proof.* We prove each property separately.

1. Let $s \neq k$. Then

$$
\begin{aligned}
\mathrm{eDur}\left((\xi|I_k), I_s\right) &= \sum_{i \in I_s} (\xi|I_k)_i \ (i - a_s) \\
&= 0.
\end{aligned}
$$

2. Let $s < k$, by Proposition 10, we then have that

$$\begin{aligned}\operatorname{Dur}\left((\xi|I_k), I_s\right) &= \operatorname{eDur}\left((\xi|I_k), I_s\right) + |I_s|\operatorname{Term}\left((\xi|I_k), \bigcup_{l=s+1}^{N} I_l\right)\\ &= \operatorname{eDur}\left((\xi|I_k), I_s\right) + |I_s|\operatorname{Term}\left(\xi, I_k\right)\\ (1) &= 0 + |I_s| \sum_{i \in I_k} \xi_i.\end{aligned}$$

3. We use the same equality as in the last property.

$$\begin{aligned}\operatorname{Dur}\left((\xi|I_k), I_s\right) &= \operatorname{eDur}\left((\xi|I_k), I_s\right) + |I_s|\operatorname{Term}\left((\xi|I_k), \bigcup_{l=k+1}^{N} I_l\right)\\ &= \operatorname{eDur}\left(\xi, I_k\right).\end{aligned}$$

We use the same equality one last time. Let $k < s$, then

$$\begin{aligned}\operatorname{Dur}\left((\xi|I_k), I_s\right) &= \operatorname{eDur}\left((\xi|I_k), I_s\right) + |I_s|\operatorname{Term}\left((\xi|I_k), \bigcup_{l=s+1}^{N} I_l\right)\\ (1) &= 0.\end{aligned}$$

□

Later in the thesis, we will also need a notion of 'the complement of duration'. For the sake of completeness, we will define this concept here as well as providing some simple properties and intuitions.

**Definition 19.** ***Duration Complement***
*The complement of the duration of a bond $\mathcal{B}(a,b)$ in a period $I \subseteq \mathbb{T}$ is the length of the intersection of the complement of the running time of the bond and $I$.*[a]

$$Dur^C\left(\mathcal{B}(a,b), I\right) := |\overline{(0,a]} \cap I| + |\overline{(b,T]} \cap I|.$$

*The complement of the duration of a bond investment $\mathcal{B}(a,b)$ in a period $I \subseteq \mathbb{T}$ is the sum of duration complements of the bonds in the investment*

$$Dur^C\left(\xi, I\right) := \sum_{i=1}^{T} \xi_i \; Dur^C\left(\mathcal{B}_i, I\right)$$

[a]Note that we take the complement in $\mathbb{T}$ so $A^C = \mathbb{T} \setminus A$

The above definition are intuitive, but we can provide equivalent definitions that are easier to work with in a mathematical context.

**Proposition 12.** *The duration complement of a bond* $\mathcal{B}(a,b)$ *in a period* $I \subseteq \mathbb{T}$ *can be equivalently defined as*

$$Dur^C\left(\mathcal{B}(a,b), I\right) = |I| - Dur\left(\mathcal{B}(a,b), I\right).$$

*Proof.* We use the following identities from set theory,

$$\begin{aligned}
(1):&\quad A \cap B = A \backslash B^C \\
(2):&\quad |A \backslash B| = |A| - |A \cap B| \\
(3):&\quad A \cap (B \cup C) = (A \cap B) \cup (A \cap C) \\
(4):&\quad A \cap B = \emptyset \implies |A \cup B| = |A| + |B|.
\end{aligned}$$

Using these identities, the proof follows from the definition.

$$\begin{aligned}
\mathrm{Dur}^C\left(\mathcal{B}(a,b), I\right) &= |\overline{(0,a]} \cap I| + |\overline{(b,T]} \cap I| \\
(1) &= |I \backslash \overline{(a,T]}| + |I \backslash \overline{(0,b]}| \\
(2) &= |I| - \left|I \cap \overline{(a,T]}\right| + |I| - \left|I \cap \overline{(0,b]}\right| \\
&= 2|I| - \left|I \cap \left(\overline{(a,b]} \cup \overline{(b,T]}\right)\right| - \left|I \cap \left(\overline{(0,a]} \cup \overline{(a,b]}\right)\right| \\
(3) &= 2|I| - \left|\left(I \cap \overline{(a,b]}\right) \cup \left(I \cap \overline{(b,T]}\right)\right| - \left|\left(I \cap \overline{(0,a]}\right) \cup \left(I \cap \overline{(a,b]}\right)\right| \\
(4) &= 2|I| - 2\left|I \cap \overline{(a,b]}\right| - \left|I \cap \overline{(0,a]}\right| - \left|I \cap \overline{(b,T]}\right| \\
(4) &= 2|I| - \left|I \cap \overline{(a,b]}\right| - \left(I \cap \left(\overline{(0,a]} \cup \overline{(a,b]} \cup \overline{(b,T]}\right)\right) \\
&= 2|I| - \mathrm{Dur}\left(\mathcal{B}(a,b), I\right) - |I| \\
&= |I| - \mathrm{Dur}\left(\mathcal{B}(a,b), I\right).
\end{aligned}$$

□

**Proposition 13.** *The duration complement of a bond investment* $\xi$ *in a period* $I \in \mathbb{T}$ *can be equivalently defined as*

$$Dur^C\left(\xi, I\right) = C|I| - Dur\left(\xi, I\right).$$

*Proof.* We again use the definition as well as the alternative definition for the duration complement of bonds proven above.

$$\begin{aligned}
\mathrm{Dur}^C(\xi, I) &= \sum_{i=1}^{T} \xi_i \ \mathrm{Dur}^C(\mathcal{B}_i, I) \\
&= \sum_{i=1}^{T} \xi_i \ \left(|I| - \mathrm{Dur}(\mathcal{B}_i, I)\right) \\
&= \sum_{i=1}^{T} \xi_i \ |I| - \sum_{i=1}^{T} \xi_i \ \mathrm{Dur}(\mathcal{B}_i, I) \\
&= C|I| - \mathrm{Dur}(\xi, I).
\end{aligned}$$

$\square$

We have so far only proven duration and termination properties of bonds without paying any mind to the interest rates of these bonds. In the next chapter, we construct a financial market upon which we impose interest rates. We will use this financial market framework to prove an important theorem that allows us to easily compare bond investments with the same distribution.

# Chapter 4

# Duration Equivalence

In this chapter, we will build up to an important theorem regarding duration, called the 'duration equivalence theorem', which is the first major result of this thesis and the theorem upon which we build the rest of the thesis. We do this by first providing some additional mathematical framework for bond investment payoffs in financial markets, which we will illustrate with an example. After providing this framework, we prove the duration equivalence theorem, upon which we discuss a number of corollary concepts. We conclude by providing some applications of duration equivalence, as well as a bound for the residual of the duration equivalence theorem.

## 4.1 Intuition behind Duration Equivalence

We illustrate the concept of duration equivalence with an example. We know that different bond investments can have the same durations. Ideally, we would like to know what the common characteristics are of all bond investments with the same durations, as this would allow us to impose these characteristics by altering the duration. The most obvious characteristic of bond investments to study is the payoff. By way of example, we create two bond portfolios both with the same durations and a normalized amount of starting cash, 1 that will be invested over a span of 100 days. We assume that all interest rates $r(i) = r$ are equal, but not known. This means that the interest rates $\rho$ of all bonds are also equal. The first portfolio $P_1$ places half of its cash into 100-day bonds and half of its cash only in overnight bonds. The second portfolio $P_2$ places all of its cash into 50-day bonds.

$$\begin{aligned}
\xi &= \frac{1}{2}\mathcal{B}_{100} + \frac{1}{2}\mathcal{B}_0 \\
\zeta &= \mathcal{B}_{50},
\end{aligned}$$

where $\mathcal{B}_0$ denotes a bond that does not fix any interest rates. Note that both portfolios have a duration of 50 and there is only one period. The interest-adjusted payoffs of both portfolios are

$$\begin{aligned}
\pi(\xi) &= \frac{1}{2}\exp\left(100\rho\right) + \frac{1}{2}\exp\left(100r\right) \\
\pi(\zeta) &= \exp\left(50\rho + 50r\right).
\end{aligned}$$

If we now assume that the interest rate is given by $r = (0.02 + 0.01J)/100$ with $J \sim \mathrm{Ber}\,(1/2)$ and so $\rho = \mathbb{E}_m[r] = 0.025/100$, then the payoffs of the bonds are

$$\begin{aligned}(\pi(\xi)|J = 0) &\approx 1.022758\\ (\pi(\zeta)|J = 0) &\approx 1.022755\\ (\pi(\xi)|J = 1) &\approx 1.027885\\ (\pi(\zeta)|J = 1) &\approx 1.027882.\end{aligned}$$

We see that in both cases $\pi(\xi)-\pi(\zeta) \approx 0.000003 = 0.3\times 10^{-6}$. So the difference is remarkably small. Intuitively, what purchasing a bond is, is removing the stochasticity of unknown interest rates by 'fixing' the interest rate to be its expectation. Purchasing a bond with $T$ days until maturity 'fixes' $T$ interest days. This means that the second portfolio fixed 50 interest days. The first portfolio however, fixed 100 days with half the money and no days with the other half. If you multiply the monetary amount placed into each bond by the amount of interest days fixed, you would again have that the second portfolio fixed 50 interest days. This leads us to believe that the bond payoff is largely dictated by the amount of interest days fixed. This is what we refer to as 'duration equivalence', as it shows that bond investment with the same duration are nearly equivalent in their payoff.[1]

**Hypothesis 1.** ***Duration Equivalence***
*If two bond investments $\xi$ and $\zeta$ have the same duration in each period of a time horizon $\mathbb{T}$, then the payoffs of $\xi$ and $\zeta$ are approximately the same.*

## 4.2 Fair Payoff in Financial Markets

In order to imitate a real financial market, we endow the time horizon $\mathbb{T}$ with various processes related to interest rates, so it becomes clear what the payout will be of bonds bought at a given time. One of the processes we want to endow $\mathbb{T}$ with is an example of a so-called prior process.

**Definition 20.** ***Prior Process***
*Let $\{u_i\}_{i\in\mathbb{T}}$ be a stochastic process over a time horizon $\mathbb{T} = \overline{(0,T]}$. A prior process of $\{u_i\}$ is a deterministic vector $u^p \in \mathbb{R}^T$ that is defined through some prior mapping $f^p : \mathcal{L}^0 \to \mathbb{R}$, where $\mathcal{L}^0$ is the set of all random variables.*

Prior functions take stochastic processes and turn them into deterministic processes. An intuitive example of a prior function is the expectation $\mathbb{E}$. This shall therefore be the prior function we apply to the interest rate process, but later on, we will define different prior mappings.

**Definition 21.** ***Interest Rate Process***
*For a time horizon $\mathbb{T}$, we define the interest rate process (IRP) to be the stochastic process $\{r_t\}_{t\in\mathbb{T}}$. Furthermore, we define the expected interest process (EIP) to be the prior process $\{\rho_t\}_{t\in\mathbb{T}}$, where $\rho_t = \mathbb{E}\,[r_t]$.*

For now, we will make no claims about the measure under which we calculate the expectation. Later on, we will distinguish two measures, so we keep in mind that the EIP process may change depending on the used

[1] Since we have defined the concept using the intuition that we want to measure the sensitivity of cash flows to the changes in interest rate, the sensitivities of bond investments with the same durations to interest rate changes are also roughly equal, which ads to the wording 'Equivalence'.

measure. Now that the time horizon is endowed with an interest rate process, we can assume that when we purchase a bond, it would be fair to receive the interest rate dictated by the IRP. This provides us with a notion of 'fair payoff'.

**Definition 22.** ***Fair Payoff***
*Let $\mathbb{T}$ be a time horizon and let $\{r_t\}_{t\in\mathbb{T}}$ be the IRP and $\{\rho_t\}_{t\in\mathbb{T}}$ the EIP of $\mathbb{T}$. Then the fair payoff of a bond purchased at time $t=0$ with running time $(a,b]$ is defined as*

$$\mathcal{B}(a,b) = exp\left(\sum_{i=1}^{a} r_i + \sum_{i=a+1}^{b} \rho_i + \sum_{i=b+1}^{T} r_i\right).$$

*The fair payoff of a bond investment is defined as the sum of the fair payoffs of the bonds within.*

If the time horizon is endowed with an IRP and EIP, then we equate the fair payoff of a bond with the interest-adjusted payoff $IAP(\mathcal{B})$. We now prove an important lemma for duration equivalence. The theorem gives a representation of the fair payoff of a bond investment.

**Proposition 14.** ***Fair Payoff of a Bond Investment***
*Let $\xi$ be a time investment over the time horizon $\mathbb{T}$ and let $\mathcal{I}$ be a time partition of $\mathbb{T}$ with $N$ members and partitioning IRP $\{R_k\}$ and partitioning EIP $\{\rho_k\}$. Then the fair payoff of $\xi$ is*

$$\pi(\xi) = \sum_{k=1}^{N}\sum_{i\in I_k} \xi_i \ exp\left(\sum_{j<k} |I_j|\rho_j + (i-a_k)\rho_k + (b_k - i)R_k + \sum_{j>k}|I_j|R_j\right),$$

*where $I_i = \overline{(a_i, b_i]}$.*

*Proof.* By definition of the payoff of a bond investment, we have that

$$\begin{aligned}\pi(\xi) &= \sum_{i=1}^{T} \xi_i \ IAP(\mathcal{B}_i)\\ &= \sum_{i=1}^{T} \xi_i \ \exp\left(\sum_{j=1}^{i} \rho_j + \sum_{j=i+1}^{T} r_j\right)\end{aligned}$$

Notice that for every term, we have that $i \in I_k$ for some $k$. We can thus split the sums in the exponent into

$$(*) = \xi_i \ \exp\left(\sum_{s<k}\sum_{j\in I_s} \rho_j + \sum_{j=a_k+1}^{i} \rho_j + \sum_{j=i+1}^{b_k} r_j + \sum_{s>k}\sum_{j\in I_s} r_j\right).$$

Now since $r_j$ is constant on every $I_s$ and therefore also $\rho_j = \mathbb{E}[r_j]$ is constant on every $I_s$, we have

$$(*) = \xi_i \ \exp\left(\sum_{s<k}|I_s|\mathbb{E}[R_s] + (i-a_k)\mathbb{E}[R_k] + (b_k-i)R_k + \sum_{s>k}|I_s|R_s\right)$$

□

This lemma acts as the last building block for proving the duration equivalence theorem, which we will prove in the next section.

## 4.3 The Duration Equivalence Theorem

Duration equivalence allows for a major reduction of the FX swap portfolio optimization problem. After proving the theorem, we will provide the reduced problem statement.

**Theorem 2.** ***Duration Equivalence***
*Let $\mathbb{T}$ be a time horizon with time partition $\mathcal{I}$, consisting of $N$ members, partitioning IRP $\{R_k\}$ and partitioning EIP $\{\rho_k\}$ so that the maximum difference of the IRP in two consecutive periods is $J_i$. Let $\xi$ and $\zeta$ be two bond investments over $\mathbb{T}$ with the same capital limit $C$. If $\forall I \in \mathcal{I} : Dur(\xi, I) = Dur(\zeta, I)$, then*

$$\pi(\xi) = \pi(\zeta) + Z,$$

*where $I_{max}$ is the longest time period and $Z = \mathcal{O}(N^2|I_{max}|^2 J_{max}^2 C)$.*

*Proof.* Let $\mathcal{I} = \{I_i\}$ and call $R_i$ the value of $\{r_t\}$ on the period $I_i$. Consider the fair payoff of a bond investment $\xi$.

$$\pi(\xi) = \sum_{k=1}^{N} \sum_{i \in I_k} \xi_i \exp\left( \sum_{j<k} |I_j|\rho_j + (i - a_k)\rho_k + (b_k - i)R_k + \sum_{j>k} |I_j|R_j \right)$$

We now focus on every time period separately. We consider the term

$$(*) = \sum_{i \in I_k} \xi_i \exp\left( \sum_{j<k} |I_j|\rho_j + (i - a_k)\rho_k + (b_k - i)R_k + \sum_{j>k} |I_j|R_j \right),$$

Applying Taylor, we get

$$(*) = \sum_{i \in I_k} \xi_i \left( 1 + \sum_{j<k} |I_j|\rho_j + (i - a_k)\rho_k + (b_k - i)R_k + \sum_{j>k} |I_j|R_j + Z_i \right),$$

where

$$\begin{aligned} Z_i &= \mathcal{O}\left( \left( \sum_{j<k} |I_j|\rho_j + (i - a_k)\rho_k + (b_k - i)R_k + \sum_{j>k} |I_j|R_j \right)^2 \right) \\ &= \mathcal{O}\left( N^2 |I_{max}|^2 J_{max}^2 \right), \end{aligned}$$

where $I_{max}$ is the longest period in the partition and $J_{max}$ is the biggest jump. The bound on the jump is due to the fact that $J_{max} \geq |\rho_i - r_i|$ for every $i$. Now letting $\Xi_k = \sum_{i \in I_k} \xi_i$, we get

$$\begin{aligned}
(*) &= \Xi_k + \sum_{j<k} \rho_j \sum_{i \in I_k} \xi_i |I_j| + \sum_{i \in I_k} (i - a_k) \xi_i \rho_k \\
&\quad + \sum_{i \in I_k} \xi_i (b_k - i) R_k + \sum_{j>k} R_j \sum_{i \in I_k} \xi_i |I_j| + \sum_{i \in I_k} \xi_i Z_i \\
&= \Xi_k + \sum_{j<k} \rho_j \mathrm{Dur}\left((\xi | I_k), I_j\right) + \mathrm{Dur}\left((\xi | I_k), I_k\right) \rho_k \\
&\quad + (\Xi_k |I_k| - \mathrm{Dur}\left((\xi | I_k), I_k\right)) R_k + \sum_{j>k} R_j \Xi_k |I_j| + \sum_{i \in I_k} \xi_i Z_i \\
&= \Xi_k + \sum_{j=1}^{N} \left[\mathrm{Dur}\left((\xi, I_k), I_j\right) \rho_j + \left(\Xi_k |I_j| - \mathrm{Dur}\left((\xi, I_k), I_j\right)\right) R_j\right] + \sum_{i \in I_k} \xi_i Z_i
\end{aligned}$$

It is trivial that $\sum_{k=1}^{N} (\xi | I_k) = \xi$. Therefore we can recombine the sum

$$\begin{aligned}
\pi(\xi) &= \sum_{k=1}^{N} \Xi_k + \sum_{j=1}^{N} \left[\mathrm{Dur}\left((\xi, I_k), I_j\right) \rho_j + \left(\Xi_k |I_j| - \mathrm{Dur}\left((\xi, I_k), I_j\right)\right) R_j\right] + + \sum_{i \in I_k} \xi_i Z_i \\
&= C + \sum_{j=1}^{N} \sum_{k=1}^{N} \left[\mathrm{Dur}\left((\xi, I_k), I_j\right) \rho_j + \left(\Xi_k |I_j| - \mathrm{Dur}\left((\xi, I_k), I_j\right)\right) R_j\right] + \sum_{i \in I_k} \xi_i Z_i \\
&= C + \sum_{j=1}^{N} \left[\mathrm{Dur}\left(\xi, I_j\right) \rho_j + \left(C |I_j| - \mathrm{Dur}\left(\xi, I_j\right)\right) R_j\right] + \sum_{i=1}^{T} \xi_i Z_i \\
&= C + \sum_{j=1}^{N} \left[\mathrm{Dur}\left(\xi, I_j\right) \rho_j + \mathrm{Dur}^C\left(\xi, I_j\right)\right] R_j + \sum_{i=1}^{T} \xi_i Z_i
\end{aligned}$$

Note that $\sum_{i=1}^{T} \xi_i Z_i \leq C \max\{Z_i\}$, so $Z \coloneqq \sum_{i=1}^{T} \xi_i Z_i = \mathcal{O}\left(C \max\{Z_i\}\right) = \mathcal{O}(N^2 |I_{max}|^2 J_{max}^2 C)$ □

Using the duration equivalence theorem, we can reformulate the main problem we aim to solve in this thesis.

**Problem Statement 2.**
*For a time partition $\mathcal{I}$, what is the optimal duration allocation?*

## 4.4 Duration Constraint

Now that the duration equivalence theorem has been proven, we will provide one of the key constraints of the duration optimization problem. In order to do this, we introduce some general notational concepts. The first of which allows us to compare set function values of sets with different sizes.

**Definition 23.** ***Averaging Operator***
*Let $\mu$ be a set function. We define the averaging operator $\bar{\cdot}$ as*

$$\overline{\mu}(S) = \frac{\mu(S)}{|S|},$$

*for any finite set $S$.*

We call $\overline{\mu}(S)$ the 'average $\mu$ of $S$'. The second notational concept, allows us to more compactly write functions applied to periods of a time partition.

**Definition 24.** ***Vectorised Notation***
*Let $f : \mathcal{I} \to S$ be some function, where $S$ is any set and $\mathcal{I} = \{I_i\}$. We denote $f(\mathcal{I})$ to be the vector*

$$f(\mathcal{I})_i = f(I_i).$$

Through this notation, we can for example write $\mathrm{Dur}\,(\xi, \mathcal{I})$ to be the duration vector of $\xi$ over the time partition $\mathcal{I}$. Using these notational conveniences, we first prove a lemma after which we show a duration vector constraint.

**Lemma 1.** *Let $\xi$ be a bond investment on the time partition $\mathcal{I} = \{I_i\}$, then*

$$\overline{Dur}\,(\xi, I_{k-1}) = \overline{Dur}\,(\xi, I_k) - \overline{eDur}\,(\xi, I_k) + \sum_{i \in I_k} \xi_i + \overline{eDur}\,(\xi, I_{k-1})$$

*Proof.* Let $\xi$ be a bond investment. By Proposition 10, we have that

$$\begin{aligned}
\overline{\mathrm{Dur}}\,(\xi, I_{k-1}) &= \overline{\mathrm{eDur}}\,(\xi, I_{k-1}) + \mathrm{Term}\left(\xi, \bigcup_{s=k}^{N} I_s\right) \\
&= \overline{\mathrm{eDur}}\,(\xi, I_{k-1}) + \mathrm{Term}\,(\xi, I_k) + \left(\mathrm{Term}\left(\xi, \bigcup_{s=k+1}^{N} I_s\right) + \overline{\mathrm{eDur}}\,(\xi, I_k)\right) - \overline{\mathrm{eDur}}\,(\xi, I_k) \\
&= \overline{\mathrm{eDur}}\,(\xi, I_{k-1}) + \sum_{i \in I_k} \xi_i + \overline{\mathrm{Dur}}\,(\xi, I_k) - \overline{\mathrm{eDur}}\,(\xi, I_k)\,.
\end{aligned}$$

□

We use this decomposition of average effective duration to prove the following theorem for buy-only bond investments.

**Theorem 3.** *Let $\xi$ be a bond investment with non-negative components over the time partition $\mathcal{I} = \{I_i\}$, then the vector $\overline{Dur}\,(\xi, \mathcal{I})$ is decreasing.*

*Proof.* We use Lemma 1 and the fact that for any $k$

$$\begin{aligned}\overline{\mathrm{eDur}}\left(\xi, I_k\right) &= \frac{1}{|I_k|}\sum_{i\in I_k}\xi_i\ (i-a_k)\\ &\leq \frac{1}{|I_k|}\sum_{i\in I_k}\xi_i\ |I_k|\\ &= \sum_{i\in I_k}\xi_i.\end{aligned}$$

So we have that

$$\begin{aligned}\overline{\mathrm{Dur}}\left(\xi, I_{k-1}\right) &= \overline{\mathrm{Dur}}\left(\xi, I_k\right) - \overline{\mathrm{eDur}}\left(\xi, I_k\right) + \sum_{i\in I_k}\xi_i + \overline{\mathrm{eDur}}\left(\xi, I_{k-1}\right)\\ &\geq \overline{\mathrm{Dur}}\left(\xi, I_k\right) + \overline{\mathrm{eDur}}\left(\xi, I_{k-1}\right).\end{aligned}$$

Now since $\xi$ only has non-negative components, we know that $\overline{\mathrm{eDur}}\left(\xi, I_{k-1}\right) \geq 0$, so the propositions holds. $\square$

The above theorem tells us that if the solution of a duration optimization problem yields an average duration vector that is increasing, then we can not construct a bond investment for that duration vector.

## 4.5 Duration Distribution

Now that we have established that all bond investment with a certain duration vector have roughly the same payoff.

**Definition 25.** ***Duration Distribution***
*Let $\mathcal{I}$ be a time partition over the time horizon $\mathbb{T}$. A duration distribution is a function $D : \mathcal{I} \to \mathbb{R}_{\geq 0}$ so that $\overline{D}$ is decreasing.*[a]

[a]Since we have a concept of ordering in $\mathcal{I}$, the fact that $D$ is increasing simply says that $D(I_k) \geq D(I_{k+1})$.

By Theorem 3, we know that for any bond investment $\xi$ and time partition $\mathcal{I}$, the duration vector $\mathrm{Dur}\left(\xi, \mathcal{I}\right)$ is a duration distribution. So now a duration distribution can be used to represent a class of bond investments with roughly the same payoff. We can represent the accuracy of this approximation by looking at the range of all payoffs of bond investments in such a class.

**Definition 26.** ***Payoff of a Duration Distribution***
*Let $D$ be a duration distributions. We define the payoff range of $D$ with capital limit $C$ as*

$$\pi(D, C) := \left[\min V, \max V\right],$$

*where $V = \{\pi(\xi) | Dur\left(\xi\right) = D, \sum \xi = C\}$.*

We will later discuss the bound on the error term in the duration equivalence theorem, during which the above definition will play a role. Much like the capital limit for bond investment, we can also put a restriction on duration distributions in order to derive additional properties.

**Definition 27.** ***Duration Restriction***
*We say that the duration distribution $D$ is restricted by $L$ if*

$$\overline{D}(I_1) \leq L.$$

*We say that $L$ is a tight restriction if $\overline{D}(I_1) = L$ and we call $D$ a normalized duration distribution if 1 is a tight restriction.*

If a bond investment is restricted by some capital limit, it makes sense that the duration distribution of that bond investment is then also in some way restricted.

**Proposition 15.** *The duration distribution of a bond investment $\xi$ is restricted by the capital limit $C$ of $\xi$.*

*Proof.* Let $C$ be the capital limit of $\xi$, then

$$\begin{aligned}\overline{\mathrm{Dur}}\,(\xi, I_1) &= \frac{1}{|I_1|}\sum_{i=1}^{T} \xi_i \,\mathrm{Dur}\,(\mathcal{B}_i, I_1)\\ &\leq \frac{1}{|I_1|}\sum_{i=1}^{T} \xi_i \,|I_1|\\ &= \sum_{i=1}^{T} \xi_i\\ &= C.\end{aligned}$$

□

We now aim to create a single representative for the payoffs of a duration distribution, given a capital limit $C$, $\pi_R(D, C)$. We know that in order to do this, this representative must have the property that

$$\pi_R(D) = C + \sum_{k=1}^{N} \left[D(I_k)\rho_j + D^C(I_k)R_k\right] + \mathcal{O}(\delta^2),$$

in order to match the last step of the proof of Theorem 2. Note here that $D^C$ corresponds to the duration complement and is thus given by $D^C(I_k) = C|I_k| - D_k$. One way to achieve this is by setting this representative equal to

$$\pi_R(D) = C\exp\left(\frac{1}{C}\sum_{k=1}^{N} D(I_k)\rho_k + D^C(I_k)R_k\right)$$

**Definition 28.** ***Flat Payoff***
*Let $\mathcal{I}$ be a time partition with $N$ periods and partitioning IRP and EIP respectively $\{R_k\}, \{\rho_k\}$. The flat payoff of a duration distribution $D$ given a restriction $L$ on $\mathcal{I}$ is defined as*

$$\pi_F(D, L) := Lexp\left(\frac{1}{L}\sum_{k=1}^{N} D(I_k)\rho_k + D^C(I_k)R_k\right),$$

*where $D^C(I_k) = L|I_k| - D(I_k)$. If no restriction is given, the tight restriction $L = \overline{D}(I_1)$ is assumed.*

Note that this flat payoff only makes sense if there is some bond investment $\xi$ that has duration distribution $D$ and capital limit $L$. We will show in the next section that such a bond investment always exists.

## 4.6 Constructing Bond Investments from Duration Distributions

It is quite easy to determine the duration distribution of a bond investment, it is however not trivial that every duration distribution has a corresponding bond investment. The latter does however turn out to be the case.

**Theorem 4.** ***Bond Investment Existence***
*For any duration distribution $D$ over a time partition $\mathcal{I}$, a bond investment $\xi$ with non-negative components, duration distribution $D$ and capital limit $\overline{D}(I_1)$ exists.*

We prove this theorem by providing the algorithm for constructing this bond investment. We call the bond investment constructed through this algorithm the vertical bond investment belonging to the given duration distribution.

**Algorithm 1.** ***Vertical Bond Investment***
*Consider the duration distribution $D : \mathcal{I} \to \mathbb{R}_{\geq 0}$. We start with the last period $I_N$, so set $k = N$ and set $\xi = \vec{0}$.*

1. *Set $E_k = \overline{D}(I_k) - \sum_{i>k} E_i$.*
2. *Set $\xi_{b_k} = E_k$, where $I_k = \overline{(a_k, b_k]}$.*

**Lemma 2.** *The vertical bond investment construction provides a bond investment with duration distribution $D$, capital limit $\overline{D}(I_1)$ and non-negative components.*

*Proof.* We show the three parts of the lemma separately.

- **Duration Distribution**

The duration of the resulting $\xi$ is given for each $I_k \in \mathcal{I}$ by

$$\begin{aligned}
\mathrm{Dur}\,(\xi, I_k) &= \sum_{i=1}^{T} \xi_i \;\mathrm{Dur}\,(\mathcal{B}_i, I_k) \\
&= \sum_{s=1}^{N} \xi_{b_s} \;\mathrm{Dur}\,(\mathcal{B}_{b_s}, I_k) \\
&= \sum_{s \geq k} E_s |I_k| \\
&= E_k |I_k| + \sum_{s>k} E_s |I_k| \\
&= |I_k| \left( \overline{D}(I_k) - \sum_{s>k} E_s \right) + \sum_{s>k} E_s |I_k| \\
&= D(I_k).
\end{aligned}$$

- **Capital Limit**
  The capital limit of the resulting $\xi$ is given by

$$\begin{aligned}
C &= \sum_{i=1}^{T} \xi_i \\
&= \sum_{k=1}^{N} \xi_{b_k} \\
&= \sum_{k=1}^{N} E_k \\
&= E_1 + \sum_{k>1}^{N} E_k \\
&= \overline{D}(I_1) - \sum_{k>1}^{N} E_k + \sum_{k>1}^{N} E_k \\
&= \overline{D}(I_1).
\end{aligned}$$

- **Non-negative Components**

The components of the resulting $\xi$ are all either 0 or equal to some $E_k$. We can rewrite $E_k$ as

$$\begin{aligned}
E_k &= \overline{D}(I_k) - \sum_{i>k} E_i \\
&= \overline{D}(I_k) - E_{k+1} - \sum_{i>k+1} E_i \\
&= \overline{D}(I_k) - \overline{D}(I_{k+1}) + \sum_{i>k+1} E_i - \sum_{i>k+1} E_i \\
&= \overline{D}(I_k) - \overline{D}(I_{k+1}) \\
&\geq 0.
\end{aligned}$$

□

Now that we have shown that any duration distribution gives rise to at least one bond investment, we can derive a relation between the flat payoff and payoff range of a duration distribution.

**Corollary 2.** *Let $D$ be a duration distribution over a time partition $\mathcal{I}$ with $N$ periods and longest period $I_{max}$. Let $\{r_k\}$ be a partitioning IRP over $\mathcal{I}$ so that consecutive interest rates are no more than $J$ apart. Then there exists a bond investment $\xi$ with duration distribution $D$ such that*

$$\pi_F(D) = \pi(\xi) + \mathcal{O}(N^2 J_{max}^2 |I_{max}|^2 \overline{D}_1)$$

*Proof.* We compute the Taylor approximation of the flat payoff of the duration $D$.

$$\begin{aligned}
\pi_F(D) &= \overline{D}_1 \exp\left(\frac{1}{\overline{D}_1} \sum_{k=1}^{N} D(I_k)\rho_k + \big(|I_k|\overline{D}_1 - D(I_k)\big) R_k\right) \\
&= \overline{D}_1 \left(1 + \frac{1}{\overline{D}_1} \sum_{k=1}^{N} D(I_k)\rho_k + (|I_k|\overline{D}_1 - D(I_k))R_k + Z\right) \\
&= \overline{D}_1 + \sum_{k=1}^{N} D(I_k)\rho_k + (|I_k|\overline{D}_1 - D(I_k))R_k + \overline{D}_1 Z,
\end{aligned}$$

where $Z$ is the Taylor remainder term, so

$$\begin{aligned}
Z &= \mathcal{O}\left(\left(\frac{1}{\overline{D}_1} \sum_{k=1}^{N} D(I_k)\rho_k + (|I_k|\overline{D}_1 - D(I_k))R_k\right)^2\right) \\
&= \mathcal{O}(N^2 J^2 |I_{max}|^2),
\end{aligned}$$

Now by Lemma 2, we know that there is a bond investment $\xi$ with capital limit $\overline{D}_1$ so that $\mathrm{Dur}\,(\xi, \mathcal{I}) = D$. By the duration equivalence theorem, we know that we can write the payoff of this bond investment as

$$\begin{aligned}
\pi(\xi) &= \overline{D}_1 + \sum_{k=1}^{N} \left[\mathrm{Dur}\,(\xi, I_k)\,\rho_k + \big(\overline{D}_1|I_k| - \mathrm{Dur}\,(\xi, I_k)\big) R_k\right] + Z^* \\
&= \overline{D}_1 + \sum_{k=1}^{N} \left[D_k\rho_k + \big(\overline{D}_1|I_k| - D_k\big) R_k\right] + Z^*
\end{aligned}$$

So we see that

$$\begin{aligned}\pi_F(D) &= \pi(\xi) - Z^* + Z \\ &= \pi(\xi) + \mathcal{O}(N^2 J_{max}^2 |I_{max}|^2 \overline{D}_1)\end{aligned}$$

□

We have now provided one construction for a bond investment with a given duration distribution. In many cases, it is possible to construct many more bond investments with the same duration distribution, but in practice such constructions are difficult to formulate for general cases. In order not to spend too much time on these constructions, we will mainly focus on the vertical bond investment construction for the remainder of this thesis. It should be noted however that the construction of bond investments may be an interesting topic for further research.

## 4.7 Applications of Duration Equivalence

We now briefly discuss some of the more direct and straight-forward applications of duration equivalence and the reduction of an FX Swap or sovereign bond portfolio to a duration distribution. We will not discuss portfolio optimization as this is thoroughly covered in chapter 6.

The intuition behind all applications of duration equivalence is that two sovereign bond portfolios with the same durations have the same payoff. In addition to this property, we have already established that the duration of a sovereign bond portfolio is a measure for the sensitivity of the portfolio value to changes in the expected interest rates for each period. This means that two portfolios with the same duration have the same payoff and risk exposures, so when an investor needs to choose between two such portfolios, these two portfolios as equal from a risk-reward standpoint. Since these two portfolios have this similar risk and reward behaviour, they can also be hedged against each other.

The hedging application can be illustrated by the example in section 4.1. In this example, the time partition was $\mathcal{I} = (0, 100)$ and the two portfolios were

$$\begin{aligned}\xi &= \frac{1}{2}\mathcal{B}_{100} + \frac{1}{2}\mathcal{B}_0 \\ \zeta &= \mathcal{B}_{50}.\end{aligned}$$

Suppose that we are trading in a market were the 100-day bonds are always the most liquid bonds and suppose that a trader has purchased a 50-day bond from someone. The investor does not want to hold on to the risk exposure from the bond, but selling it will require crossing the order book spread, which is more expensive for 50-day bonds than for 100-day bonds. The duration equivalence theorem now says that instead of crossing the larger spread, the investor may choose to purchase 100-day bonds with half of the money and put the rest of the money in a money market fund (assuming the money-market fund has the sovereign interest rate). The resulting cash flows cancel each other, so the 50-day bond is now hedged.

The above hedging example can of course be extended to FX swaps and is only one example of the ability for duration equivalence to simplify risk exposures of sovereign bond and FX swap portfolios. Through the new concept of duration and duration equivalence theorem, duration can be traded in the same way as deltas for options. This simplifies the intuitions of sovereign bond and FX swap trading and makes it easier to assess portfolio positions and hedge undesired exposures.

## 4.8 Duration Equivalence Error

We finish off this chapter with some more computationally heavy derivations to see if we can find favourable bounds for the remainder term of the duration equivalence theorem. We know that the error grows with the a second order trend in the amount of periods, the maximum period length and the jump size, but if one is willing to go through some more computational steps, more tangible bounds can be obtained.

**Theorem 5.** ***Duration Equivalence Error***
*Let $\xi$ and $\zeta$ be two bond investments with the same duration distribution and capital limit $C$ and let the conditions of 2 be met. Let $R_0$ be the initial interest rate and let $B_j$ be the bound on the interest difference $|R_j - R_0|$. Then*

$$|\pi(\xi) - \pi(\zeta)| \leq \frac{1}{2}\sum_{i=1}^{T} |\xi_i - \zeta_i|\theta^2 e^{\theta},$$

*where*

$$\theta = \sum_{j=1}^{N} |I_j|(|R_0| + B_j).$$

*Proof.* In the proof of the duration equivalence theorem, we calculate the first-order tailor term around the point $x_0 = 0$. This results in the following equality.

$$e^x = 1 + x + \frac{e^y}{2}x^2,$$

for some $y$ in between 0 and $x$. We first attempt to bound $|x|$, where

$$\begin{aligned}
|x| &= \left| \sum_{j<k} |I_j|\rho_j + (i - a_k)\rho_k + (b_k - i)R_k + \sum_{j>k} |I_j| R_j \right| \\
\text{(Triangle ineq.)} &\leq \sum_{j<k} |I_j||\rho_j| + (i - a_k)|\rho_k| + (b_k - i)|R_k| + \sum_{j>k} |I_j||R_j|
\end{aligned}$$

Now since $|R_j - R_0| \leq B_j$, we have that

$$\begin{aligned}
|R_j| &= |R_j - R_0 + R_0| \\
&\leq |R_j - R_0| + |R_0| \\
&\leq B_j + |R_0|
\end{aligned}$$

And since $|R_j| < B_j + |R_0|$, we have that also $|\rho_j| = |\mathbb{E}[R_j]| < B_j + |R_0|$, therefore

$$\begin{aligned}
|x| &\leq \sum_{j<k} |I_j||\rho_j| + (i-a_k)|\rho_k| + (b_k - i)|R_k| + \sum_{j>k} |I_j||R_j| \\
&=\leq \sum_{j<k} |I_j|(B_j + |R_0|) + (i-a_k)(B_j + |R_0|) + (b_k - i)(B_j + |R_0|) + \sum_{j>k} |I_j|(B_j + |R_0|) \\
&= \sum_{j<k} |I_j|(B_j + |R_0|) + (b_k - a_k)(B_j + |R_0|) + \sum_{j>k} |I_j|(B_j + |R_0|) \\
&= \sum_{j=1}^{N} |I_j|(|R_0| + B_j),
\end{aligned}$$

which is equal to the defined $\theta$. The rest term can now be bound by

$$\frac{e^y}{2}x^2 \leq \frac{e^y}{2}\theta^2.$$

Now since the exponential function is increasing, plugging in the maximum number for $y$ provides the maximum error. If $x$ is positive, then $y \leq x = |x| \leq \theta$. If $x$ is negative, $y \leq 0 \leq |x| \leq \theta$, so

$$\frac{e^y}{2}x^2 \leq \frac{1}{2}\theta^2 \exp(\theta)$$

Now recall from the proof of the duration equivalence theorem that we can write

$$\pi(\xi) = C + \sum_{j=1}^{N} \left[\mathrm{Dur}(\xi, I_j)\,\rho_j + (C|I_j| - \mathrm{Dur}(\xi, I_j))\,R_j\right] + \sum_{i=1}^{T} \xi_i Z_i,$$

where now the $Z_i$ can be bound by $\frac{1}{2}\theta^2 \exp(\theta)$. We therefore have that since the durations of $\xi$ and $\zeta$ are the same,

$$\begin{aligned}
|\pi(\xi) - \pi(\zeta)| &= \left|\sum_{i=1}^{T} \xi_i Z_i - \sum_{i=1}^{T} \zeta_i Z_i\right| \\
&= \left|\sum_{i=1}^{T} (\xi_i - \zeta_i) Z_i\right| \\
\text{(Triangle ineq.)} &\leq \sum_{i=1}^{T} |\xi_i - \zeta_i||Z_i| \\
&\leq \frac{1}{2}\sum_{i=1}^{T} |\xi_i - \zeta_i|\theta^2 \exp(\theta)
\end{aligned}$$

□

Since we have proven a general form of the theorem, we can easily prove some additional corollaries that are useful in practice.

**Corollary 3.** ***Duration Equivalence Error***

1. *The duration equivalence error is bound by*

$$|\pi(\xi) - \pi(\zeta)| \leq C\theta^2 e^\theta.$$

2. *The term for theta is bound by*

$$\theta \leq \sum_{j=1}^{N} |I_j|(|R_0| + jJ_{max})$$

*Proof.* The second result is trivial as $B_j \leq jJ_{max}$ by definition of $J_{max}$. For the first assertion, we have

$$\begin{aligned} \sum_{i=1}^{T} |\xi_i - \zeta_i| &\leq \sum_{i=1}^{T} \xi_i + \sum_{i=1}^{T} \zeta_i \\ &= 2C. \end{aligned}$$

□

This above can be used a simplified form of Theorem 5, as it does not require as many known variables. As the last theorem of this chapter, we derive a bound for the error of the flat payoff.

**Theorem 6.** ***Flat Payoff Error***
*Let $\xi$ be a bond investment with duration distribution $D$ and capital limit $C$. Assume the same conditions as Theorem 5. Then*

$$|\pi_F(D, C) - \pi(\xi)| \leq C\theta^2 e^\theta,$$

*where*

$$\theta = \sum_{k=1}^{N} |I_k|(|R_0| + B_k).$$

*Proof.* We now focus on the Taylor remainder term of the first-order Taylor expansion derived in Corollary 2. This remainder term has the same form as in the duration equivalence error proof as it is also the Taylor expansion of a polynomial, $\frac{1}{2}x^2e^y$ but this time, we plug in

$$x = \sum_{k=1}^{N} D_k\rho_k + (|I_k|C - D_k)R_k$$

We can use similar steps as in the proof of the duration equivalence error to obtain

$$
\begin{aligned}
|x| &\leq \sum_{k=1}^{N} D_k(|R_0| + B_k) + (|I_k|C - D_k)(R_0 + B_k) \\
&= C \sum_{k=1}^{N} |I_k|(R_0 + B_k)
\end{aligned}
$$

The rest of the proof is the same as the duration equivalence error derivation in combination with Corollary 3. $\square$

The obtained bound can now be used to link the payoff range of a duration distribution with the flat payoff.

**Corollary 4.** ***Payoff of a Duration Distribution***
*Let $D$ be a duration distribution and let $\pi(D, C)$ be the payoff range of $D$ with capital limit $C$, then*

$$\pi(D, C) \subseteq [\pi_F(D, C) - C\theta^2 e^{\theta}, \pi_F(D, C) + C\theta^2 e^{\theta}],$$

*where*

$$\theta = \sum_{k=1}^{N} |I_k|(R_0 + B_k).$$

# Chapter 5

# Risk and Return Modelling

Now that we have established duration equivalence, the remainder of this thesis is concerned with a major application of this theorem: Portfolio Optimization. The duration equivalence theorem, can be used to formulate an FX Swap portfolio optimization problem as a duration optimization problem, which we will call the 'Duration Allocation Problem'. Before defining this problem, we need a notion of risk and expected return, as these play a major role in most portfolio optimization problem. For both components, a model is required to calculate risk and reward parameters based on available market information. The framework we will use for our portfolio optimization is the Markowitz framework [6]. In this framework, the reward component is quantified by a vector whose components each correspond to the expected return of one asset in the portfolio and the risk component is quantified by the covariance matrix of the assets. In our duration model, the 'assets' are the interest rates in each period. In this chapter, we will model how we can use market information to model returns. After that, we propose various models for the covariance matrix and conclude by selecting the most suitable model for the remainder of the thesis.

## 5.1 Return Modelling

Before jumping into modelling the return, we first note that since we are more interested in the risk component of FX swap portfolios than the return, we can also choose not to regard the return component at all. The advantage of this is that we are able to create the absolutely least risky portfolio, but the downside is that we may have to pay a disproportionate premium for this low risk. In practice, traders are often able to make predictions on where the market is going based on recent market events. These predictions, even if they only slightly differ from the market expectations, provide valuable insight into the risk-reward payoff of FX swap portfolios and we will thus be taking them into account when possible.

When trading on the exchange, a trader has information on what the price of financial products are, which corresponds to what the market thinks these products are worth. In the case of FX swaps, the asset being traded is essentially the interest rate difference of the two currencies that are being swapped, and the prices reflect what the market expects these interest rates to do. In practice, the market may be wrong and there may be some other, better estimate for what the interest rate will do.

**Definition 29.** ***Interest Estimator***
*An interest estimator $\beta$ on a time partition $\mathcal{I}$ with partitioning IRP $\{R_k\}$ is a partitioning prior process where $\beta = \mathbb{E}_b[R_k]$, where $\mathbb{E}_b$ is the expectation under some chosen measure.*

We now make the distinction between the EIP $\rho_k = \mathbb{E}_m[R_k]$, where $\mathbb{E}_m$ denotes the market expectation and therefore the market price of the interest rate and $\beta_k = \mathbb{E}_b[R_k]$, where $\mathbb{E}_b$ denotes the expectation of the interest rate based on some empirical measure. This interest estimator is a function of available data.

For regular assets, the return is modelled to be the quotient of the expected value of the asset and the cost of the asset. Applying this way of defining returns to bond investing, we get

$$P = \frac{\pi(\xi|\rho)}{\pi(\xi|\beta)},$$

Where we use $\pi(\xi|\beta)$ to denote the payoff of $\xi$ is we use $\beta$ as the prior process. Since we know that the payoff of any $\xi$ with the same duration can be represented by the flat payoff of the duration, we can take the representative

$$\begin{aligned} P &= \frac{\pi_F(D|\rho)}{\pi_F(D|\beta)} \\ &= \exp\left(\frac{1}{L}\left(D \cdot \rho + F \cdot R\right) - \frac{1}{L}\left(D \cdot \beta + F \cdot R\right)\right) \\ &= \exp\left(\frac{1}{L}\left(D \cdot (\rho - \beta)\right)\right), \end{aligned}$$

where $L$ is the a tight restriction of $D$ and $F_i = L|I_i| - D(I_i)$. This formula for the quotient return motivates the construction of some return metrics.

**Definition 30.** ***Return of a Duration Distribution***
*Let $\mathcal{I}$ be a time partition with IRP $\{R_k\}$, EIP $\{\rho_k\}$ and IB $\{\beta_k\}$. We define the flat log return of a duration distribution $D$ with tight restriction $L$ to be*

$$W_F(D) \coloneqq \frac{1}{L} D \cdot (\rho - \beta).$$

Note that since the exponential function is an increasing function, a higher flat log return also yields a higher return. Since the flat log return is a linear function, it is more convenient to use in optimization problems, so this will be our return component.

## 5.2 Interest Rate Modelling

Now that we have determined how to model the return component, we want to say something about the risk component of the objective function. In order to do this, we first need to formulate a model for the interest rates. To do this, we use a very simple mode which allows interest rates to jump up, jump down and remain the same, with a fixed jump size. This behaviour is very similar to a Bernoulli distribution, but with a third component.

**Definition 31.** ***Signed Bernoulli Distribution***
*We say that $X$ is a signed Bernoulli random variable with up-parameter $p$ and down-parameter $q$, or $X \sim sBer(p, q)$ if*

$$\begin{aligned}\mathbb{P}(X = 1) &= p\\ \mathbb{P}(X = 0) &= 1 - p - q\\ \mathbb{P}(X = -1) &= q.\end{aligned}$$

If the current interest rate is known, then the subsequent interest rate is simply equal to the current one plus the potential jump.

**Definition 32.** ***Interest Jump Model***
*We say that a stochastic process $\{R_k\}$ follows an interest jump model (IJM) with base interest $R_b$ and jump size $J$ if*

$$\begin{cases} R_1 = R_b, \\ R_{k+1} = R_k + X_k J, \quad k \geq 1, \end{cases}$$

*where $X_k \sim sBer(p_k, q_k)$. $\{X_k\}$ is called the underlying jump process.*

Because of the simple nature of the signed Bernoulli distribution, we can already derive a number of properties of these distributions, as well as the interest jump model.

**Proposition 16.** ***Properties of the Signed Bernoulli Distribution***
*Let $X \sim sBer(p, q)$, then the following properties hold.*

1. *The expectation of $X$ is given by*

$$\mathbb{E}[X] = p - q$$

2. *The variance of $X$ is given by*

$$Var(X) = p + q - (p - q)^2$$

3. *If $q = 0$, then $X \sim Ber(p)$ and if $p = 0$, then $-X \sim Ber(q)$.*

*Proof.* Both 1 and 3 follow immediately from the definition, for 2, we have

$$\begin{aligned}\mathrm{Var}(X) &= \mathbb{E}\left[X^2\right] - \mathbb{E}[X]^2\\ &= p + q - (p - q)^2\end{aligned}$$

□

**Theorem 7.** ***Interest Jump Covariance***
*Let $\{R_k\}$ follow an IJM with base interest $R_b$ and jump size $J$. If $m < n$, then the covariance of $R_m$ and $R_n$ is given by*

$$
\begin{aligned}
Cov(R_m, R_n) &= Var(R_m) + J\,Cov\left(R_m, \sum_{i=m}^{n-1} X_i\right) \\
&= J^2 \sum_{j=1}^{m-1} Var(X_j) + J^2 \sum_{j=1}^{m-1} \sum_{i=m}^{n-1} Cov(X_j, X_i)
\end{aligned}
$$

*where $\{X_k\}$ is the underlying jump process of $\{R_k\}$.*

*Proof.* From the definition, it is clear that

$$R_n = R_b + J \sum_{i=1}^{n-1} X_i = R_m + J \sum_{i=m}^{n-1} X_i.$$

Applying linearity of the covariance now yields the desired result. □

The jump covariance theorem says that if we know the covariance matrix of the underlying jump process, then we can derive the covariance matrix of the interest rates. In order to determine the jump covariances, we need to calibrate the underlying jump process. In practice, it is often the case that information regarding interest rate jumps is available through various sources. Bloomberg has a feature for example that shows exactly how many jumps the market is pricing in per central bank meeting for various currencies.

**Definition 33.** ***Jump Expectation Process (JEP)***
*Consider a IJM with underlying jump process $\{X_k\}$. The jump expectation process of the IJM is the process $\{\eta_k\}$ so that*

$$\mathbb{E}[X_k] = \eta_k - \eta_{k-1} \in [-1, 1],$$

*where we denote $\eta_0 = 0$.*

We always assume that when calibrating a jump process, a JEP is available. In the next section we will see that for some simple variants of the interest jump model, this information is sufficient for a full calibration, but for more involved variants, some additional assumptions need to be made.

## 5.3 Special Interest Jump Models

We now define some special cases of interest jump models that have some additional properties. These additional properties allows us to make more inferences about covariance properties and in some cases can vastly simplify the model.

An intuitive way to simplify the interest jump model is by only allowing a jump in one direction. This makes sense as when the market expects 0.8 jumps for a meeting, the market deems it highly likely that given the current state of the world, the central bank would want to increase the interest rate. Given this information, it would be very odd for the central bank to decrease the interest rate, so the underlying jump distribution is more likely to be $\text{sBer}(0.8, 0)$ than for instance $\text{sBer}(0.9, 0.1)$.

**Definition 34.** ***Simplified Interest Jump Model***
*A simplified interest jump model (SIJM) is an interest jump model where the underlying jump process is of the form $X_k \sim sBer(p_k, 0)$ or $X_k \sim sBer(0, q_k)$.*

The underlying jump distribution are then in essence regular Bernoulli random variables. If there is a period where the conditions for interest rate increases are met for a longer time period, it can be that the market only expects these increases for a longer time.

**Definition 35.** ***Directional Jump Expectation Process (DJEP)***
*We say that a jump expectation process is up-directional if*

$$\eta_i - \eta_{i-1} \geq 0.$$

*We say that a JEP is down-directional if*

$$\eta_i - \eta_{i-1} \leq 0.$$

When calibrating an underlying jump process using such a directional jump expectation, it makes sense to use a simplified model.

**Definition 36.** ***Directional Interest Jump Model***
*A directional interest jump model (DIJM) is a model where the underlying jump process is either of the form $X_k \sim sBer(p_k, 0)$ (up-directional) or of the form $X_k \sim sBer(0, q_k)$ (down-directional).*

We saw that in the case of FX swaps, the cost[1] of the swap is dependent on the difference of the two interest rates of the currencies that are swapped. If the meetings of the underlying central banks do not occur on the same day, an increase in one interest rate and then in the other interest rate causes the FX swap price to temporarily go up or down, but then go back to its original positions short after.

**Definition 37.** ***Alternating Jump Expectation Process (AJEP)***
*We say that a jump expectation process $\{\eta_k\}$ is alternating if there exists positive sequence $\{\nu_k\}$ such that*

$$\eta_k - \eta_{k-1} = (-1)^{k-1}\nu_k.$$

Here, the initial jump is upward, this can be inverted by considering the process $\{-\eta_k\}$. We can again model such behaviour using a simplified jump model.

[1] Again, this refers to the premium paid, which can be negative or positive, hence it is possible for the cost to be negative.

**Definition 38.** ***Alternating Interest Jump Model***
*An alternating interest jump model (AIJM) is an IJM where the underlying jump process is of the form*

$$X_k \sim sBer(p_k, q_k), k > 1,$$

*where* $p_{2k} = 0$ *and* $q_{2k-1} = 0$ *for each* $k$*.*

## 5.4 Covariance Modelling

We now proceed to model the covariances of interest rate jumps. We will start by modelling independent interest rate jumps and deriving the properties of such a model. We will then proceed to modelling Markovian interest rate jumps, which we will see make for a significantly more complicated covariance structure, even when restricting ourselves to simplified jump models. We will conclude by manually imposing a correlation structure to the model and deriving the corresponding covariance structure.

We note that in this section, we will define logical concepts and derive simple results for them in order to tell a coherent and logically structured story. In the context of FX swap trading, not all discussed concepts are directly applicable and will therefore not be further discussed or evaluated in this thesis, but omitting these concepts would result in a confusing, incoherent story.

### 5.4.1 Independent Interest Rate Jumps

The simplest covariance structure we can impose on the interest jump model is independence of the jumps.

**Definition 39.** ***Independent Interest Jump Model (IIJM)***
*An independent interest jump model (IIJM) is an IJM such that the underlying jump process consists of pairwise independent random variables.*

In this case, Theorem 7 gives us a formula for the covariance matrix.

**Corollary 5.** *The covariance matrix of an IIJM* $\{R_k\}$ *is*

$$\Sigma(R) = Diag(Var(R))$$

Placing this restriction on the model also allows us to fully characterise all underlying jump models when a jump expectation process is provided.

**Proposition 17.** ***Independent Jump Characterization***
*Let $\eta$ be a JEP. The class of all IJMs with JEP $\eta$ and independent underlying jump processes is characterised by*

$$X_k \sim sBer(p_k, q_k),$$

*choosing $p_k$ and $q_k$ so that $\eta_k = \eta_{k-1} + p_k - q_k, \eta_1 = p_1 - q_1$.*

*Proof.* The covariance condition clearly holds because all $X_k$ are pairwise independent. The expectation condition holds because

$$\begin{aligned}\eta_k &= \eta_{k-1} + p_k - q_k \\ &= \eta_1 + \sum_{i=2}^{k}(p_k - q_k) \\ &= \sum_{i=1}^{k}(p_k - q_k) \\ &= \sum_{i=1}^{k}\mathbb{E}\left[X_k\right].\end{aligned}$$

□

When placing additional restriction on the model, this characterisation provides us with small, at times even unique underlying jump processes, which are easy to study.

**Corollary 6.** *Let $\eta \in R^N$ be a jump expectation process, then there exists a unique SIJM for $\eta$ with independent jumps. If $\eta$ is up-directional, then there exists a unique up-directional IJM for $\eta$ with independent jumps. If $\eta$ is down-directional, then there exists a unique down-directional IJM for $\eta$ with independent jumps. If $\eta$ is alternating, then there exists a unique AIJM for $\eta$ with independent jumps. These processes are respectively given by*

$$\begin{aligned}X_k^s &\sim sBer\left(\delta_k^+, \delta_k^-\right) \\ X_k^u &\sim sBer(\eta_k - \eta_{k-1}, 0) \\ X_k^d &\sim sBer(0, \eta_{k-1} - \eta_k) \\ X_k^a &\sim sBer(p_k, q_k),\end{aligned}$$

*where $\delta_k^+ = \max(\eta_k - \eta_{k-1}, 0), \delta_k^- = \max(\eta_{k-1} - \eta_k, 0)$, $p_k = (\eta_k - \eta_{k-1})\mathbb{1}(k/2 \notin \mathbb{N})$ and $q_k = (\eta_k - \eta_{k-1})\mathbb{1}(k/2 \in \mathbb{N})$.*

### 5.4.2 Markovian Jump Processes

We now proceed to construct interest jump models where the next jump depends on the previous jump and only on the previous jump. We will only construct two such processes, but of course many different such processes exist. Since allowing this Markovian property creates many degrees of freedom to construct an IJM with, we place heavy restrictions on the resulting Markov process.

The first restriction we place is that we only consider simplified interest jump models, reducing the signed Bernoulli distributions to regular Bernoulli distribution. If we further restrict ourselves to monotone jump expectation processes, we can already define a jump process. These two restrictions vastly reduce the degrees of freedom we have, but there are still many possibilities. To really hone in on a single process, we make the following two assumptions: If the market prices in more than one jump in the next two meeting and the next meeting does not have a jump, then the subsequent meeting definitely has a jump. If the market prices in less than one jump in the next two meetings and the next meeting does have a jump, then the subsequent meeting will definitely not have a jump. These two assumptions, combined with the other two constraints, give rise to the Duration Markov Jump process.

**Proposition 18.** ***Directional Markov Jumps (DMJ)***
*Let $\eta$ be an up-directional JEP. We construct the directional Markov jump (DMJ) process as follows.*

$$\begin{aligned} X_1 &\sim sBer(\eta_1, 0) \\ X_k &\sim \begin{cases} sBer(A_k, 0), & \eta_k - \eta_{k-2} \le 1 \\ sBer(B_k, 0), & \eta_k - \eta_{k-2} > 1, \end{cases} \end{aligned}$$

*where $\eta_0 = 0$ and*

$$\begin{aligned} A_k &= (1 - X_{k-1}) \frac{\eta_k - \eta_{k-1}}{1 - \eta_{k-1} + \eta_{k-2}} \\ B_k &= X_{k-1} \frac{\eta_k - \eta_{k-2} - 1}{\eta_{k-1} - \eta_{k-2}} + (1 - X_{k-1}). \end{aligned}$$

*Then $\mathbb{E}[X_k] = \eta_k - \eta_{k-1}$, and $\{X_k\}$ is up-directional.*

*Proof.* We use induction to prove the claim. The base case clearly holds as $\mathbb{E}[X_1] = \eta_1$. We now progress using induction

- Suppose $\eta_k - \eta_{k-2} \le 1$. Then $X_k \sim \text{sBer}(A_k, 0)$ and choosing $p_{k-1}$ such that $X_{k-1} \sim \text{sBer}(p_{k-1}, 0)$,

$$\begin{aligned} \mathbb{E}[X_k] &= \mathbb{P}(X_{k-1} = 0)\mathbb{E}[X_k | X_{k-1} = 0] + \mathbb{P}(X_{k-1} = 1)\mathbb{E}[X_k | X_{k-1} = 1] \\ &= (1 - p_{k-1}) \frac{\eta_k - \eta_{k-1}}{1 - \eta_{k-1} + \eta_{k-2}} + p_{k-1} \cdot 0 \\ &= (1 - \mathbb{E}[X_{k-1}]) \frac{\eta_k - \eta_{k-1}}{1 - \mathbb{E}[X_{k-1}]} \\ &= \eta_k - \eta_{k-1} \end{aligned}$$

- Suppose $\eta_k - \eta_{k-2} > 1$. Then $X_k \sim \text{sBer}\left(B_k, 0\right)$ and choosing $p_{k-1}$ such that $X_{k-1} \sim \text{sBer}\left(p_{k-1}, 0\right)$,

$$\begin{aligned}
\mathbb{E}\left[X_k\right] &= \mathbb{P}(X_{k-1}=0)\mathbb{E}\left[X_k \middle| X_{k-1}=0\right] + \mathbb{P}(X_{k-1}=1)\mathbb{E}\left[X_k \middle| X_{k-1}=1\right] \\
&= (1-p_{k-1}) \cdot 1 + p_{k-1}\frac{\eta_k - \eta_{k-2} - 1}{\eta_{k-1} - \eta_{k-2}} \\
&= (1 - \mathbb{E}\left[X_{k-1}\right]) + \mathbb{E}\left[X_{k-1}\right]\frac{\eta_k - \eta_{k-2} - 1}{\mathbb{E}_{k-1}} \\
&= \eta_k - \eta_{k-2} - \mathbb{E}\left[X_{k-1}\right] \\
&= \eta_k - \eta_{k-1}.
\end{aligned}$$

□

A similar construction can be made for down-directional jump expectation vectors. Since the dependence structure of Markov processes is simple and the Bernoulli distribution also is a very simple distribution, we can derive the covariance between Markov sequence elements.

**Theorem 8.** ***DMJ Covariance***
*The covariance of two consecutive elements of a DMJ process is*

$$Cov\left(X_k, X_{k-1}\right) = -\mathbb{E}\left[X_k\right]\mathbb{E}\left[X_{k-1}\right],$$

*if* $\eta_k - \eta_{k-2} \leq 1$ *and*

$$Cov\left(X_k, X_{k-1}\right) = -\left(1 - \mathbb{E}\left[X_k\right]\right)\left(1 - \mathbb{E}\left[X_{k-1}\right]\right),$$

*if* $\eta_k - \eta_{k-2} > 1$.

*Proof.* We distinguish two cases

- Suppose $\eta_k - \eta_{k-2} \leq 1$, then

$$\begin{aligned}
\text{Cov}\left(X_k, X_{k-1}\right) &= \mathbb{E}\left[X_k X_{k-1}\right] - \mathbb{E}\left[X_k\right]\mathbb{E}\left[X_{k-1}\right] \\
&= \underbrace{\mathbb{P}(X_k = 1 | X_{k-1} = 1)}_{=0}\mathbb{P}(X_{k-1}=1) - \mathbb{E}\left[X_k\right]\mathbb{E}\left[X_{k-1}\right] \\
&= -\mathbb{E}\left[X_k\right]\mathbb{E}\left[X_{k-1}\right] \\
&= -(\eta_k - \eta_{k-1})(\eta_{k-1} - \eta_{k-2}).
\end{aligned}$$

- Suppose $\eta_k - \eta_{k-2} > 1$, then

$$\begin{aligned}
\text{Cov}\left(X_k, X_{k-1}\right) &= \mathbb{E}\left[X_k X_{k-1}\right] - \mathbb{E}\left[X_k\right]\mathbb{E}\left[X_{k-1}\right] \\
&= \mathbb{P}(X_k = 1 | X_{k-1} = 1)\mathbb{P}(X_{k-1}=1) - \mathbb{E}\left[X_k\right]\mathbb{E}\left[X_{k-1}\right] \\
&= \frac{\eta_k - \eta_{k-2} - 1}{\eta_{k-1} - \eta_{k-2}}\mathbb{E}\left[X_{k-1}\right] - (\eta_k - \eta_{k-1})(\eta_{k-1} - \eta_{k-2}) \\
&= \mathbb{E}\left[X_k\right] + \mathbb{E}\left[X_{k-1}\right] - 1 - \mathbb{E}\left[X_k\right]\mathbb{E}\left[X_{k-1}\right].
\end{aligned}$$

□

We see that for consecutive sequence elements the calculations already become quite extensive. Calculating covariances of all elements in the sequence is thus not realistic.[2] We do observe however, that the covariance of consecutive elements is negative. This matches our expectation, as we said that in both cases, one outcome for the previous jump increases the likelihood of the other outcome for the next jump. We can thus create a truncated covariance matrix.

**Definition 40.** ***Damp-Truncated Alternating Covariance***
*Let $X$ be a random vector so that the covariance of consecutive components is given by $Cov(X_{i-1}, X_i) = c_i$. We define the damp-truncated alternating covariance structure with damping factor $\alpha$ to be.*

$$\Sigma^t(X_i, X_j) \coloneqq c_j \left(\frac{-1}{\alpha}\right)^{k-1},$$

*where $k = j - i > 0$. and $\Sigma^t(X_i, X_i) \coloneqq Var(X_i)$.*

Note that this damp-truncated alternating covariance structure can be used to truncate the covariance matrix of any sequence where consecutive sequence elements are negatively correlated with a decreasing correlation trend. We now proceed to change the monotonicity constraint to instead require the jump expectation process to be alternating. This constraints simulates the behaviour of the difference of two interest rates with correlated behaviour. In this case if one of the interest rates jumps up, the other is likely to follow, causing the difference to go back down. We thus assume the same negatively correlated behaviour as before, but this time, we need to formulate a more nuanced correlation structure. Instead of saying with certainty that a jump will occur given some condition, we simply say that a subsequent opposite jump is more likely to occur.

**Theorem 9.** ***Alternating Markov Jumps (AMJ)***
*Let $\eta$ be an alternating JEP. We construct the alternating Markov jump (AMJ) process with amplification factor $\phi$ as follows.*

$$\begin{aligned} X_1 &\sim sBer(\eta_1, 0), \\ X_{2k} &\sim sBer(0, A_{2k}) \\ X_{2k+1} &\sim sBer(A_{2k+1}, 0), \end{aligned}$$

*where*

$$A_k = \frac{-(-1)^k \delta_k}{1 + (\phi - 1)\delta_{k-1}}(1 + (\phi - 1)|X_{k-1}|).$$

*In the above equality, $\delta_k = \eta_k - \eta_{k-1}$. The sequence $\{X_k\}$ is alternating and $\mathbb{E}[X_k] = \eta_k - \eta_{k-1}$.*

*Proof.* We use induction to prove the claim. The base case trivially holds. We prove the claim for the even

[2]Ideally, one could inductively prove a general formula for covariance. This will likely prove to be challenging as every induction step will be dependent on $\eta_k - \eta_{k-2}$, creating many different cases. Further research to special cases and heuristic may yield interesting and useful results, though this is outside the scope of this thesis.

iterations. The odd iterations follow a similar logic.

$$\begin{aligned}\mathbb{E}\left[X_{2k}\right] &= \mathbb{E}\left[X_{2k} \mid X_{2k-1}=1\right] \mathbb{P}(X_{2k-1}=1) + \mathbb{E}\left[X_{2k} \mid X_{2k-1}=0\right] \mathbb{P}(X_{2k-1}=0) \\ &= \mathbb{E}\left[X_{2k} \mid X_{2k-1}=1\right] \mathbb{E}\left[X_{2k-1}\right] + \mathbb{E}\left[X_{2k} \mid X_{2k-1}=0\right] \left(1-\mathbb{E}\left[X_{2k-1}\right]\right) \\ &= \frac{-\delta_{2k}\phi}{1+(\phi-1)\delta_{2k-1}}\delta_{2k-1} + \frac{-\delta_{2k}}{1+(\phi-1)\delta_{2k-1}}(1-\delta_{2k-1}) \\ &= \frac{-\delta_{2k}\left(\phi\delta_{2k-1}-\delta_{2k-1}+1\right)}{1+(\phi-1)\delta_{2k-1}} \\ &= \delta_{2k}.\end{aligned}$$

□

The intuition for the amplification factor is that after a jump in one currency, a jump in the other currency becomes $\phi$ times as likely. One criticism of the AMJ model is that this amplification factor between jumps does not carry over. This means that it is possible for the difference in the two interest rates to grow over time. We can again make a claim about the covariance of consecutive sequence elements.

**Theorem 10.** ***AMJ Covariance***
*The covariance of two consecutive elements of an AMJ process is*

$$Cov(X_k, X_{k-1}) = \delta_k\delta_{k-1}\left(\frac{\phi}{1+(\phi-1)\delta_{k-1}}-1\right).$$

*Proof.* We consider an even element and the element before it.

$$\begin{aligned}\operatorname{Cov}\left(X_{2k}, X_{2k-1}\right) &= \mathbb{E}\left[X_{2k}X_{2k-1}\right] - \mathbb{E}\left[X_{2k}\right]\mathbb{E}\left[X_{2k-1}\right] \\ &= -\mathbb{P}(X_{2k}=-1|X_{2k-1}=1)\mathbb{P}(X_{2k-1}=1) - \delta_{2k}\delta_{2k-1} \\ &= \frac{\delta_{2k}\phi}{1+(\phi-1)\delta_{2k-1}}\delta_{2k-1} - \delta_{2k}\delta_{2k-1} \\ &= \delta_{2k}\delta_{2k-1}\left(\frac{\phi}{1+(\phi-1)\delta_{2k-1}}-1\right).\end{aligned}$$

We now look at an even element and the element after it.

$$\begin{aligned}\operatorname{Cov}\left(X_{2k}, X_{2k+1}\right) &= \mathbb{E}\left[X_{2k}X_{2k+1}\right] - \mathbb{E}\left[X_{2k}\right]\mathbb{E}\left[X_{2k+1}\right] \\ &= -\mathbb{P}(X_{2k+1}=1|X_{2k}=-1)\mathbb{P}(X_{2k}=-1) - \delta_{2k}\delta_{2k+1} \\ &= \frac{\delta_{2k+1}\phi}{1+(\phi-1)\delta_{2k}}\delta_{2k} - \delta_{2k}\delta_{2k+1} \\ &= \delta_{2k}\delta_{2k+1}\left(\frac{\phi}{1+(\phi-1)\delta_{2k}}-1\right).\end{aligned}$$

□

Here again, we see that the calculations for consecutive sequence elements are already becoming quite involved. Because we do not have different cases this time, calculating covariances of other sequence elements is possible and may yield interesting results in some further research. Since our main objective of this thesis is not covariance modelling, however, we will settle for again using the damp-truncated alternating covariance structure.

### 5.4.3 Correlation Structure

We have now constructed a covariance structure for independent and Markovian jump processes. For the Markovian processes, we already saw that we had to set some very strict constraints on the jump process to make any meaningful claims about the covariance. In the case were such constraint are not suitable, making general claims about the covariance is thus infeasible.

In the event that an investor would want to optimize a fixed income portfolio without applying these constraints, the investor may choose to manually define a covariance structure for the jump process. Doing this is quite difficult, as covariance is not a universally well-understood concept. To reduce covariance to a more intuitive concept, we recall the following covariance property.

**Proposition 19.** ***Covariance Decomposition***
*Let $X$ be a random vector and let $\Sigma$ be its covariance matrix. We can decompose $\Sigma$ into*

$$\Sigma = Diag\left(\sigma_X\right) K_X Diag\left(\sigma_X\right),$$

*where $\sigma_X$ is the vector with standard deviations of $X$ and $K_X$ is the correlation matrix of $X$.*

The proof of the above theorem follows from the definition of correlation. While covariance is not always intuitive, everybody has an intuition for what the correlation between two random variables is. An investor may construct the correlation matrix of the jump process and derive the covariance matrix since the standard deviations are already known. A more in-depth intuition of how correlation between Bernoulli random variables works is given in Appendix C.

## 5.5 Model Selection

As we stated in the introduction to the previous section, we will not use or evaluate all of the discussed covariance models. In order to not needlessly complicate the covariance modelling, we will restrict ourselves to using the Alternating Markovian Jump (AMJ) model in combination with the Damp-Truncated Alternating Covariance with damping factor $\alpha = 2$.

We will not dive into a parameter analysis of the damping factor or compare the formulated covariance structures as this falls well outside the scope of this thesis and the choice of parameters and models depends on factors such as application, market conditions an risk appetite. All of the stated models are easily implemented so an interested reader can perform a more in-depth analysis with relative ease depending on the use-case. This last remark especially holds for manually imposed correlation structure, which would be very difficult to apply larger-scale tests to.

# Chapter 6

# Portfolio Optimization

In this section we formulate the duration allocation problem, which we base on the Markowitz Model, a frequently applied method to optimize portfolios. After formulating the problem, we rewrite it to more closely match the classic Markowitz model. We conclude by deriving a number of properties of the duration allocation problem.

## 6.1 The Markowitz Model and Markowitz Bullet

As previously stated, we model the duration allocation problem after the so-called Markowitz model. We refer to [6] for the formulation and solutions of this model.

**Definition 41.** ***Markowitz Optimization Problem***
*The standard Markowitz optimization problem is defined as*

$$\begin{cases} \min \; x^T \Sigma x \\ x^T M = m \\ \langle x, 1_d \rangle = 1 \\ x \in \mathbb{R}^d, \end{cases}$$

*where $\Sigma$ is a covariance matrix.*

When working with the Markowitz model, one is often interested in solving the optimization problem not just for a single $m$, but for a larger range, so that an investor can decide whether the risk-reward payoff for a given $m$ is worth it. It thus becomes interesting to look at the solution in function of $m$.

**Definition 42.** ***Markowitz Bullet***
*We denote the set of solutions for the Markowitz optimization problem for any possible $m$ by $\mathcal{F}$. The geometric shape in $\mathbb{R}^2$ defined by $\left\{(x^T \Sigma x, x^T M) | x \in \mathcal{F}\right\}$ is called the Markowitz Bullet.*

For a standard Markowitz optimization problem, the solution set is known.

**Theorem 11.** ***Solution to the Markowitz Problem***
*If $\Sigma$ is invertible, then the solution set to the Markowitz Optimization problem is given by*

$$\mathcal{F} = \{x_a + \lambda \nu_{a,b}, \lambda \in \mathbb{R}\},$$

*with* $a = ||1_d||_{\Sigma^{-1}}$, $b = \langle M, 1_d \rangle_{\Sigma^{-1}}$,

$$x_a = \frac{1}{a}\Sigma^{-1}1_d \qquad \nu_{a,b} = \frac{\Sigma^{-1}(M - \frac{b}{a}1_d)}{||M - \frac{b}{a}1_d||_{\Sigma^{-1}}}.$$

We clearly see that in order for this solution set to be sensible, we need $\Sigma$ to be invertible, which can be seen as an additional constraint.

## 6.2 Formulating the Duration Allocation Problem

We now construct the duration optimization problem using the Markowitz framework. We start by formulating the objective function. The risk in the duration portfolio is not caused by the amount of interest rate days that are fixed, but rather by the ones that are not. This means the risk of the portfolio is

$$(L||\mathcal{I}|| - D)^T \Sigma (L||\mathcal{I}|| - D) = F^T \Sigma F,$$

where $L$ is the tight restriction of $D$. We immediately see that if we express the optimization problem in terms of $D$, we are left with an undesirable quadratic form. Instead, we choose to formulate the optimization problem in terms of $F$, the duration complement. The return constrained is modelled using the flat log return defined in Definition 30, which is defined as $\frac{1}{L}D^T(\rho - \beta)$. For the sake of simplicity, we normalise the duration distribution, so $L = 1$. The expected return of the portfolio is then given by

$$\exp\left(D^T M\right) = \exp\left(\left(||\mathcal{I}|| - F\right)^T M\right).$$

The reward constraint is thus given by

$$\left(||\mathcal{I}|| - F\right)^T M = \ln(m) \implies F^T M = ||\mathcal{I}||^T M - \ln(m).$$

Since $m$ is simple a parameter for the model, we can set $m^* = \ln(m)$ to solve the optimization problem. We then simply need to remember to translate the resulting Markowitz bullet to account for this transformation. We thus see that we have a quadratic objective function and a linear return constraint, both of which match the Markowitz model. We now however recall the remaining constraints of the duration distribution.

$$\overline{D} \text{ decreasing} \implies \overline{F} \text{ increasing},$$
$$\overline{D} \in [0,1]^d \implies \overline{F} \in [0,1]^d.$$

We now add the additional constraint that $\overline{F}_d = 1$. It will later become clear why we add this constraint.

**Definition 43.** ***Duration Allocation Problem***
*We define the duration allocation problem as the optimization problem.*

$$\begin{cases} \min \ F^T \Sigma F \\ F^T M = ||\mathcal{I}||^T M - m \\ \overline{F} \ \textit{increasing} \\ \overline{F}_d = 1. \end{cases}$$

These constraint do not nicely match the Markowitz constraints. Particularly the first of these constraints is particularly undesirable. In the next section, we aim to reformulate the optimization problem to require more desirable constraints.

## 6.3 Optimization Constraints

We first discuss the monotonicity constraint of $\overline{F}$. We do this by considering the difference between consecutive elements of $\overline{F}$. We define the operator $\Delta\overline{F} \in \mathbb{R}^d$ such that

$$(\Delta\overline{F})_i = \overline{F}_i - \overline{F}_{i-1},$$

where we define $\overline{F}_0 = 0$. We now have that

$$\Delta\overline{F} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ -1 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & -1 & 1 \end{bmatrix} \overline{F}$$

So we see that the operator $\Delta$ corresponds to a matrix transformation. Furthermore, this matrix transformation is invertible with its inverse being the lower-triangular matrix with only ones on the lower triangle.

**Definition 44.** ***Difference Operator and Cumulative Sum Operator***
*We define the difference operator in $d$ dimensions to be $\Delta = TriDiag(-1, 1, 0) \in \mathbb{R}^d$ and we define the cumulative sum operator in $d$ dimensions to be the lower-triangular matrix $K \in \mathbb{R}^d$ with only ones on the lower triangle.*

**Proposition 20.** *The difference operator and cumulative sum operator are each other's inverse.*

$$\Delta^{-1} = K$$

We now observe that if a vector is increasing, the difference of consecutive component is always positive, so

$$\overline{F} \text{ increasing} \iff \Delta\overline{F} \succeq 0,$$

where $\vec{x} \succeq 0 \iff \forall i : x_i \geq 0$. This constraint is a lot easier to work with than the monotonicity constraint. We therefore wish to express the optimization problem in terms of $\Delta\overline{F}$. Now since $\Delta$ is simply a matrix operation, it is possible to change between $\overline{F}$ and $\Delta\overline{F}$ in the optimization problem, but changing from $F$ to $\overline{F}$ is only possible if the operator $\bar{\cdot}$ is also a matrix transformation.

**Proposition 21.** ***Averaging Operator***
*Let $\mu$ be a set function on $S$, a finite, ordered set of sets and let $u = \mu(S)$. The averaging operator $\bar{\cdot}$ can be written as*

$$\overline{u} = G^{-1}u,$$

*where $G = Diag\,(||S||)$.*

We thus see that $\Delta\overline{F} = \Delta G^{-1}F$, where $G = \text{Diag}\,(||\mathcal{I}||)$. We now define $V = \Delta\overline{F}$ and rewrite the optimization problem in terms of $V$.

$$\begin{cases} \min\ F^T\Sigma F \\ F^T M = ||\mathcal{I}||^T M - m \\ \overline{F} \text{ increasing} \\ \overline{F}_d = 1 \end{cases}$$

$$\implies \begin{cases} \min\ (GKV)^T\Sigma(GKV) \\ (GKV)^T M = ||\mathcal{I}||^T M - m \\ V \succeq 0 \\ \sum V_i = 1 \end{cases}$$

$$\implies \begin{cases} \min\ ||V||_{K^TG\Sigma GK} \\ V^T((GK)^T M) = ||\mathcal{I}||^T M - m \\ V \succeq 0 \\ \langle V, 1_d \rangle = 1. \end{cases}$$

So we see that we not only resolved the monotonicity constraint, but we also obtain the constraint that the solution corresponds to a weighting of assets. Note also that since both $G$ and $K$ are invertible, $\Sigma GK$ is also invertible.

The resulting optimization problem is now simply a standard Markowitz problem with an added positivity constraint. This means that we can not solve this problem analytically, but instead require computational methods. The resulting optimum can be calculated using the `qpsolvers` package in Python. Now where the standard Markowitz problem can be solved as long as $\Sigma$ is invertible, the extra constraint requires $\Sigma$ to be positive definite. The following proposition states that this condition is equivalent to $\Sigma$ being positive definite.

**Proposition 22.** *Let $Q$ be a positive definite matrix and let $P$ be an invertible matrix. Then $P^TQP$ is positive definite.*

*Proof.* Let $x \in \mathbb{R}^d$. Since $P$ is invertible, we can write $x$ as $P^{-1}y$ for some $y \in \mathbb{R}^d$, so

$$\begin{aligned} x^T P^T QPX &= y^T (P^{-1})^T PQPP^{-1} y \\ &= y^T Q y \\ &> 0. \end{aligned}$$

$\square$

**Theorem 12.** ***Duration Allocation Problem***
*The duration allocation problem can be written as a classic long-only Markowitz problem*

$$\begin{cases} \min \; V^T \Sigma^* V \\ V^T M^* = m^* \\ \langle V, 1_d \rangle = 1 \\ V \succeq 0, \end{cases}$$

*where* $\Sigma^* = ||\Sigma||_{GK}$ *is positive definite,* $M^* = (GK^T)M$*,* $m^* = ||\mathcal{I}||^T M - m$ *and* $V = \Delta \overline{F}$*.*

In the next section, we explore conditions for existence of an optimum. We henceforth assume $\Sigma$ to be positive definite.

## 6.4 Solvability Conditions

In order to optimize an objective function subject to a set of constraints, we first need to know if there even exist values that satisfy the constraints. Such values are called feasible solutions. If we call the set of all feasible solutions $\mathcal{P}$, we can write the optimization problem as

$$\begin{cases} \min \; ||F||_\Sigma \\ F \in \mathcal{P}. \end{cases}$$

In the case of our optimization problem, the set $\mathcal{P}$ is given by

$$\mathcal{P}(m) = \left\{ F \in \mathbb{R}^d | (||\mathcal{I}|| - F) \cdot M = m, \overline{F} \text{ increasing}, \overline{F}_d = 1 \right\}$$

Since the optimization problem formulated in the previous section is equivalent to this problem, the feasible set of the problem in terms of $V$ is the same as $\mathcal{P}$ where the necessary transformation is performed to convert $F$ into $V$.

An optimization problem can not be solved if the feasible set is empty. Note that since our feasible set depends on the parameter $m$, we must choose $m$ in such a way that the set is not empty, we call the set of all such $m$ the solvability set $\mathcal{M}$.

$$\mathcal{M} := \{ m \in \mathbb{R} | \mathcal{P}(m) \neq \emptyset \}$$

We first prove that $0 \in \mathcal{M}$.

**Proposition 23.** $\mathcal{M}$ ***is not empty***

$$0 \in \mathcal{M}$$

*Proof.* The solution corresponding to $\overline{F} = (1, 1, \ldots, 1)$ is in $\mathcal{P}(0)$. $\square$

As it turns out, we can provide a characterisation for all $m$ for which the problem is solvable. To prove this characterisation, we first prove a couple of lemmas.

**Lemma 3.** ***Convexity of the Solvability set***
*Let $M_d = 0$, then*

$$0 < m \in \mathcal{M} \implies [0, m] \subseteq \mathcal{M}$$
$$0 > m \in \mathcal{M} \implies [m, 0] \subseteq \mathcal{M}.$$

*Proof.* Suppose that $M_d = 0$ and $0 < m \in \mathcal{M}$. Let $D = ||\mathcal{I}|| - F$ and let $\tilde{D}$ and $\tilde{M}$ be the vector containing all but the last component of $D$ and $M$ respectively (so the operator $\tilde{\cdot} : \mathbb{R}^d \to \mathbb{R}^{d-1}$ projects a vector to its first $d-1$ components). Then

$$\tilde{D} \cdot \tilde{M} = D \cdot M = m,$$

since $M_d = 0$. Now let $\lambda \in [0, 1]$ and

$$F' = \begin{bmatrix} ||\tilde{\mathcal{I}}|| - \lambda(||\tilde{\mathcal{I}}|| - \tilde{F}) \\ |I_d| \end{bmatrix}$$

We then have that

$$\begin{aligned} (||\mathcal{I}|| - F') \cdot M &= \lambda(\tilde{||\mathcal{I}||} - \tilde{F}) \cdot \tilde{M} + |I_d| M_d \\ &= \lambda D \cdot M \\ &= \lambda m. \end{aligned}$$

So and $m' \in [0, m]$ invokes a nonempty feasible set with $F' \in \mathcal{P}(m)$ as long as $F'$ satisfies the other constraints. Indeed, we have that $F'_d = 1$ and since $\tilde{F} \preceq 1$ and is increasing, we also have that $F' \preceq 1$ and is increasing. $\square$

We now proceed to provide a lemma surrounding duration distributions that either have full duration in a period or no duration.

**Definition 45.** ***Fully Concentrated Duration Distribution***
*We say that a normalized duration distribution vector is fully concentrated if $D \in \{0, 1\}^d$. The first zero-element of the duration distribution is called the turning point. We denote the set of all fully concentrated duration distributions by $\mathcal{C}$.*[a]

[a]Note that in $d$ dimensions, $\mathcal{C}$ has $d + 1$ elements.

The complement of a duration distribution is called fully concentrated if the corresponding duration distribution is fully concentrated.

**Lemma 4.** *There exist fully concentrated duration distribution complements $F_{min}$ and $F_{max}$ such that*

$$F_{min} \in \mathcal{P}(\min \mathcal{M}) \qquad\qquad F_{max} \in \mathcal{P}(\max \mathcal{M}).$$

*Proof.* We prove the claim only for $F_{max}$, the other follows the same procedure. Let $m = \max \mathcal{M}$ and suppose that $\mathcal{P}(m) \neq \emptyset$. Now let $F \in \mathcal{P}(m)$ be a non-fully concentrated duration distribution complement. Now let $\overline{F}_a$ be the first nonzero component of $\overline{F}$ and $\overline{F}_b$ be the last non-one component. We now prove that

$$\sum_{i=a}^{b}(1-\overline{F}_i)|I_i|M_i = 0.$$

We show this by contradiction. Suppose that $\sum_{i=a}^{b}(1-\overline{F}_i)|I_i|M_i = H \neq 0$. If $H < 0$, then setting $\overline{F}_i = 0$ for each $i \in \{a, \ldots, b\}$ yields a more optimal $m$. Lastly, suppose that $H > 0$, now for each $i \in \{a, b\}$, let

$$\overline{F}_i^* = 1 - \frac{1}{\overline{F}_b}\left(1 - \overline{F}\right).$$

We then have that

$$\begin{aligned}\sum_{i=a}^{b}(1-\overline{F}_i^*)|I_i|M_i &= \sum_{i=a}^{b}\frac{1}{\overline{F}_b}(1-\overline{F}_i)|I_i|M_i \\ &= \frac{1}{\overline{F}_b}H \\ &> H,\end{aligned}$$

which again yields a more optimal $m$.

Now that we have proven that $\sum_{i=a}^{b}(1-\overline{F}_i)|I_i|M_i = 0$, we observe that the duration distribution complement $F_{max}$ given by

$$F_{max}^i = \begin{cases} F_i, & i \notin \{a, \ldots, b\} \\ 1, & i \in \{a, \ldots, b\} \end{cases}$$

has the same $m$ and thus $F_{max} \in \mathcal{P}(\min \mathcal{M}) \cap \mathcal{C}$, □

Note that since the $m$ corresponds to the amount of profit for the corresponding duration allocation, the above theorem states that both the minimal and maximum profit can be obtained with fully concentrated duration distributions. This above lemma is the last building block we need to provide the characterization for the solvability set.

**Theorem 13.** ***Characterisation of the Solvability Set***
*Define the following vector*

$$X_n = \sum_{i=1}^{n} |I_i| M_i$$

*and* $X_0 = 0$*. If* $M_d = 0$*, then the solvability set* $\mathcal{M}$ *is given by*

$$\mathcal{M} = \left[\min_{n\geq 0} X_n, \max_{n\geq 0} X_n\right].$$

*Proof.* By Lemma 4, we know that there are fully concentrated duration distribution complements $F_{min}$ and $F_{max}$ that correspond to the lower and upper bounds of $\mathcal{M}$ respectively. Now for any fully concentrated duration distribution complement $F^*$ with turning point $n$, we have that

$$(||\mathcal{I}|| - F^*) \cdot M = \sum_{i=1}^{n-1} |I_i| M_i.$$

We thus have that

$$\min \mathcal{M} = \min_{F^* \in \mathcal{C}} \{(||\mathcal{I}|| - F^*) \cdot M\} = \min_{n\geq 0} X_n$$
$$\max \mathcal{M} = \max_{F^* \in \mathcal{C}} \{(||\mathcal{I}|| - F^*) \cdot M\} = \max_{n\geq 0} X_n.$$

Now by the convexity of the solvability set, we have that

$$\mathcal{M} = \left[\min_{n\geq 0} X_n, \max_{n\geq 0} X_n\right].$$

□

This theorem provides us with perfect insight on the solvability of the duration allocation problem. With the above solvability characterization, we can use the `qpsolvers` library in python to construct the the full Markowitz bullet for the duration allocation problem. Note that the solvability criterion states that under the given condition, the optimization problem is solvable for any desired profit in between the maximum attainable profit and the minimum attainable profit.

## 6.5 Concentration Limits

Since we have strayed from the analytical solvability of the Markowitz problem anyway, we can now also take the liberty of introducing additional constraints. The constraint we would like to add is a concentration limit, which prohibits the portfolio from having too many FX swaps expire in the same period. The difference in duration between consecutive period is an intuitive measure for the amount of terminations in that period.

**Definition 46.** ***Concentration Limit***
*We say that a duration allocation complement $F$ has concentration limit $L \in \mathbb{R}^d_{\geq 0}$ if*

$$\forall i : \Delta\overline{F} \preceq L.$$

If the value of a component of $V$ is high, that means many bonds have expired in the corresponding period. This means, if we want to restrict the amount of bonds expiring in a time period, we can do this by adding the constraint.

$$V \preceq L,$$

where $L$ is a vector containing the concentration limit for each period. The `qpsolvers` library has no problems handling this constraint. Note that by adding a concentration constraint, the characterization of the solvability set no longer holds.[1]

[1] A new characterization could be derived, but this derivation is not required and is thus left outside the scope of this thesis.

# Chapter 7

# Duration Allocation Results

Now that we have formulated the duration allocation problem and shown that it can be written as a classic long-only Markowitz problem, We proceed to show how the mathematical framework we constructed can be used to construct an optimal FX Swap portfolio. Note that even though we have shown that our results can be generalised to sovereign bonds, we will restrict ourselves to FX swaps in this chapter, as this was the main goal of this thesis. After showing the example, we will measure the performance of the method on simulated markets, after which we will go into more detail on how to assess the model performance when not all market data is available.

## 7.1 Applying Duration Allocation

We will now combine the results of this thesis to apply the duration allocation framework and the duration equivalence theorem to a real-world example. We first explain the exact setup of the example, after which we will calculate the optimal solution. We will then translate the duration allocation result into an FX swap portfolio. We conclude by applying the duration equivalence theorem to validate the result.

### 7.1.1 Model Inputs

The example we will use will utilize market data of the 21st of May 2024. The data we use is obtained from the WIRP feature in Bloomberg. This feature shows how many interest rate jumps the market prices in for various central bank meetings. The following meetings for the ECB and the FED were available. The values correspond to the cumulative interest rate jumps priced in by the market up to and including that meeting.

| ECB (2024) | |
|---|---|
| 06-Jun | -0.957 |
| 18-Jul | -1.164 |
| 12-Sep | -1.857 |
| 17-Oct | -2.133 |
| 12-Dec | -2.720 |

| FED (2024) | |
|---|---|
| 12-Jun | -0.026 |
| 31-Jul | -0.233 |
| 18-Aug | -0.745 |
| 07-Nov | -1.084 |
| 18-Dec | -1.652 |

Table 7.1: Meetings and priced-in interest jumps of the European Central Bank and the Federal Reserve. All dates

We see that the meetings from the two banks alternate. We combine the market data into a single table.

| Date | Nr. Days | Bank | Jump Expectation | Cumulative Jumps | Estimator |
|---|---|---|---|---|---|
| 21-May | 0 | - | 0 | 0 | 0 |
| 06-Jun | 15 | ECB | -0.957 | -0.957 | -1 |
| 12-Jun | 21 | FED | -0.026 | -0.931 | -1 |
| 18-Jul | 57 | ECB | -1.164 | -1.138 | -1 |
| 31-Jul | 70 | FED | -0.233 | -0.931 | -1 |
| 18-Aug | 113 | ECB | -0.745 | -1.624 | -2 |
| 12-Sep | 119 | FED | -1.857 | -1.112 | -1 |
| 17-Oct | 148 | ECB | -2.133 | -1.388 | -1 |
| 07-Nov | 169 | FED | -1.084 | -1.049 | -1 |
| 12-Dec | 204 | ECB | -2.720 | -1.636 | -2 |
| 18-Dec | 210 | FED | -1.652 | -1.068 | -1 |

Table 7.2: Table of all meetings, prices in jumps and jump estimators. Note that for the cumulative jumps, the USD interest jumps need to be inverted since $\Delta r = r_{€} - r_{\$}$. The initial interest rate difference on the start date is $-1.5\%$ and the jump size is 0.25%.

Since the focus of this thesis is on the mathematical framework and not a study of estimators, we use a very simple estimator. Since after a meeting either a jump occurs or does not occur, we simply choose the option that the market deems most likely. Since the market only prices in negative jumps, we assume positive jumps can not occur. The 'Nr. Days' column corresponds to the amount of days between the given meeting and today. A visual representation of these example interest rates can be found in Figure 7.1. The green-shaded region in the figure corresponds to periods where the market expects higher interest rates than the estimator. This means we should purchase swaps, as we expect to get more return than what we think the fair return is. The red-shaded region corresponds to periods where we think FX swaps will perform worse than if we do not purchase swaps.

### 7.1.2 Solving the Duration Allocation Problem

The solutions to this example problem can be found in Figure 7.2. The x-axis of this plot shows the variance of a given optimal duration allocation and the y-axis shows the return. In the plot, we show 4 results. The first is the standard duration allocation problem. The second is a duration allocation where we impose a concentration limit of 30% per period. the third result is the 3-month benchmark, which corresponds to an FX swap portfolio containing only 3-month FX swaps. The last result is the spread benchmark, which purchases the same amount of FX swaps for each day in the time horizon.[1]

The first observation we make is that the concentration-limited solution provides less return options and always has more risk than the standard duration allocation curve. We mainly focus on the concentration-limited curve as this is more widely applicable. We see that in this case, the optimal risk also corresponds to the optimal return, which means there is what we call an 'absolute optimum', which will always be the preferred portfolio. Note that in different example, this absolute optimum may not exist and so the risk-reward payoff needs to

[1] The spread benchmark is not interesting in practice as purchasing such a swap portfolio is not practical, so this benchmark is for theoretical purposes only. The 3-month benchmark is a more applicable benchmark as the 3-month swap tenor is the most liquid.

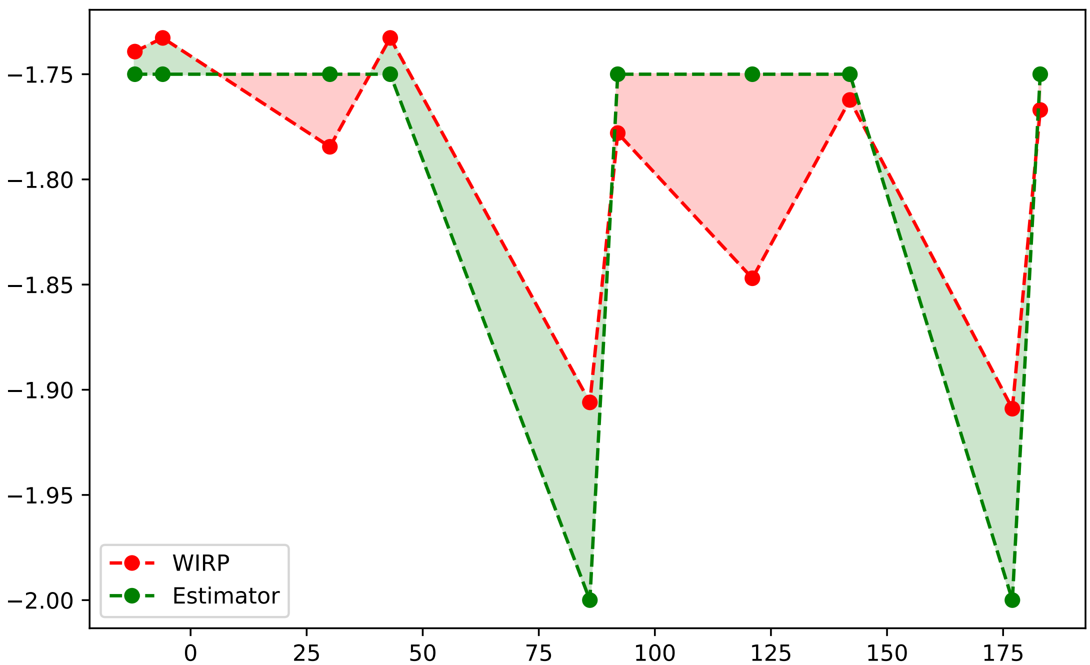


Figure 7.1: A visual representation of the expected interest rate jumps. The green line corresponds with the interest estimator and the red line with the market expectation.

be evaluated. We also see that we outperform the 3-month benchmark both in the risk and the reward sense and we outperform the spread benchmark in the risk sense, but slightly under-perform in the reward sense. Table 7.3 shows the comparison of the two benchmarks to the absolute optimal portfolio. We also calculate the risks and the rewards in the case that we would like to invest €500 million into FX swaps, as this is a realistic amount to trade in a single day. Note that in the table, we convert the variances on the x-axis to standard deviations.

| | Optimal Risk | Optimal Reward |
|---|---|---|
| 3M Benchmark | €455 257 | € − 10 458 |
| Spread Benchmark | €359 017 | €12 038 |
| Duration Allocation | €29 117 | €44 495 |
| Concentration Limited | €180 945 | €13 704 |

Table 7.3: Risk and Reward of the Various Portfolios

We see here that the standard duration allocation problem performs significantly better than all other

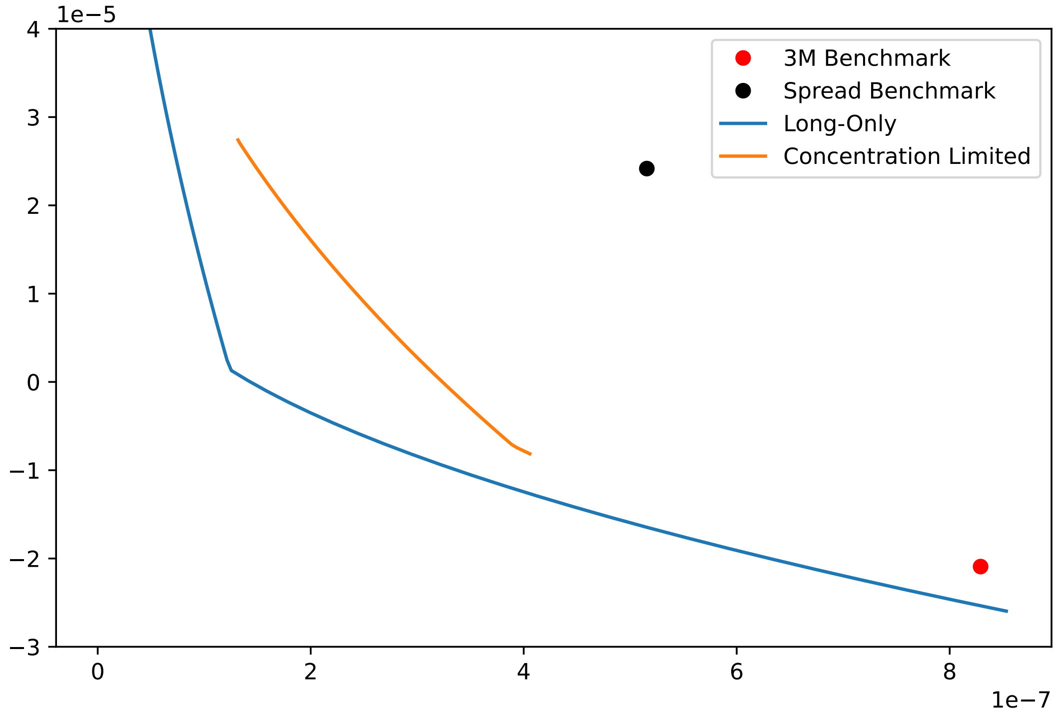


Figure 7.2: The Markowitz bullets for the example duration allocation problems

portfolio, but even when imposing the concentration limits, the duration allocation problem still reduces the risk by roughly 50% at minimum. Now that we have obtained the optimal duration allocations, we still need to translate the allocation into an FX swap portfolio. In this step, we will immediately see one of the key reasons why one would impose a concentration limit.

### 7.1.3 Translating the Duration Allocation to an FX Swap Portfolio

The corresponding duration distributions of the optimal duration allocation can be found in Table 7.4. We immediately see here that when not imposing any concentration limit, the model may be inclined to switch abruptly from full-duration to zero-duration. This switch would be fine if we are 100% certain of the model inputs, but this abrupt change makes the model very sensitive to any errors. In addition to this, in the name of safe investing, investing fully into a single FX swap tenor is not responsible, which is why certain asset managers are mandated not to do this. The concentration limited solution reduces some of the sensitivity to error and results in a more responsible investment decision that abides by potential mandates.

| Period | Duration Allocation | Concentration Limited |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 1 | 1 |
| 3 | 1 | 1 |
| 4 | 1 | 0.93 |
| 5 | 1 | 0.9 |
| 6 | 1 | 0.9 |
| 7 | 1 | 0.9 |
| 8 | 1 | 0.6 |
| 9 | 0.99 | 0.3 |
| 10 | 0 | 0 |

Table 7.4: Optimal duration allocations with and without the concentration limits

In section 4.6, we provided an algorithm for constructing an FX swap portfolio from a duration distribution. If we apply this algorithm here, we get the FX swap portfolio shown in Table 7.5. In this table, we show how many of each swap we need to purchase for the optimal portfolio. We also show which major tenor is closest to the theoretically optimal swap, as the non-standard tenors are not always available for a favorable price.

| No limit | | | Concentration limit | | |
|---|---|---|---|---|---|
| Days | Tenor | Volume | Days | Tenor | Volume |
| 168 | 6-Month | 0.01 | 56 | 2-Month | 0.073 |
| 203 | 6/7-Month | 0.99 | 69 | 2/3-Month | 0.027 |
| | | | 147 | 5-Month | 0.3 |
| | | | 168 | 5/6-Month | 0.3 |
| | | | 203 | 7-Month | 0.3 |

Table 7.5: Optimal FX swap portfolios with and without concentration limits

Note that when purchasing the standard tenors instead of the exact tenors, the duration of the resulting portfolio may change. This means the risk and reward exposures of the portfolio may deviate from the optimum. One can choose to try to deviate from the exact tenors in such a way that the optimal duration is roughly maintained, but the since the risk and reward difference will be slight, one may also choose to ignore this difference.

### 7.1.4 Duration Equivalence

The optimization of the duration allocation uses the flat log return, which is derived from the flat payoff of a duration distribution. This means we optimize the flat payoff of a duration distribution in a market with the interest rate differential $\rho - \beta$. The translation step to an FX swap portfolio can then be justified by Theorem 6. This theorem shows that the difference between the optimal duration distribution $D$ and the corresponding vertical FX swap portfolio we constructed is given by[2]

[2]Note that we take the capital limit $C = 1$, when multiplying the FX swap portfolio by 500 million, the error scales linearly as shown by the theorem.

$$|\pi_F(D) - \pi(\xi)| \leq \theta^2 e^{\theta},$$

where

$$\theta = \sum_{k=1}^{N} |I_k|(|R_0| + J_{max} j).$$

Now we can fill in the values for $|I_k|$ by the time between meetings, $R_0 = -1.5\%$ and $J_{max} = 0.25\%$. Note that $N = 10$. If we fill in these numbers, we get that

$$|\pi_F(D) - \pi(\xi)| \leq 2.431\text{e} - 4.$$

The error between the risk and the reward of the duration allocation thus differs from the risk and the reward of the given FX swap portfolios by at most the number above. In Table 7.6, we show the results from Table 7.3 accounting for this error by showing the risk and reward ranges.

| | Risk (Low) | Risk (High) | Reward (Low) | Reward (High) |
|---|---|---|---|---|
| 3M Benchmark | €455 146 | €455 368 | € − 10 611 | −10 605 |
| Spread Benchmark | €358 929 | €359 104 | €11 946 | €11 953 |
| Duration Allocation | €29 110 | €29 125 | €44 335 | €44 356 |
| Concentration Limited | €180 900 | €180 989 | €13 656 | €13 662 |

Table 7.6: Risk and Reward Ranges of the Various Portfolios

We see that the range for the risks and the rewards is very tight. No more than €220 for the risk and no more than €25 for the reward. We note here that for the risk comparison, we made a very crude translation. The risk of an investment corresponds to the standard deviation of the return. For the translation, we assume that the risk corresponds to a quantity of money that is realistic to be lost. In this context, we can simply translate this amount of money directly. In reality, other transformations may be more accurate, but for the sake of this result, this simple translation is good enough. Table 7.6 clearly shows that the risk-reward results obtained from the duration allocation problem can be translated to a nearly identical risk-reward profile in the corresponding FX swap portfolio.

## 7.2 Performance in Simulated Markets

In the previous section, we applied our method to a single example and saw that the duration allocation method outperforms the 3-month benchmark in both risk and reward. In practice, this may of course not always be the case, so we now proceed to validate the model. Generally for model validation, two very common approaches is to use historical data or simulated data and use these to measure the performance of a model. We will only validate the model using simulated data for two reasons.

- For a very long time, both the EUR and the USD interest rates were constant and low, as can be seen in Figure 2.1. This means that most of the data available is not representative for the current market and therefore any statistical inferences based on this period of time is not currently useful.

- The WIRP function in Bloomberg prices in the next to 5 FED and the next 5 ECB meetings. This is 10 meetings for which there are only a few possible outcomes, so simulating every possible market by approximation is doable.

We will thus validate the model by measuring its performance in a vast number of markets. This collection aims to be a representative for all possible markets.

Before simulating the market, we first note an important property of the duration allocation problem that reduces the number of simulations we need to make. If we multiply any predicted jumps by $-1$, intuitively, the correlation structure between jumps does not change so $\Sigma$ does not change. For the return, what happens is that $M \mapsto -M$. This means that if we solve the duration allocation problem for some market expectation $\rho$ and some interest estimator $\beta$, the optimal duration allocation is the same as the optimal duration allocation of the pair $(-\rho, -\beta)$ with the same risk and opposite return. This means if we simulate a market with the pair $(\rho, \beta)$, we can immediately infer the result of $(-\rho, -\beta)$.

We now split the simulation results into 3 parts. We first simulate as many markets as possible and assess the performance of the duration allocation solution compared to the 3-month benchmark. We then consider some lighter simulation to compare the model to some other benchmarks and infer some empirical properties of the simulations.

### 7.2.1 Basic Simulation

For the market simulations, we assume a time partition similar to the one in the example,

$$\mathcal{I} = (15, 25, 55, 65, 95, 105, 135, 145, 175).$$

Furthermore, we assume an initial interest difference of 1.5% and an interest jump of 0.25%. We multiply all results by €500 million to make the results more intuitive. For the market simulations, we assume that for every meeting, the market can expect one of 4 things:

- Jump is very unlikely: $\rho = 0.1, \beta = 0$.
- Unsure, but leaning to no jump: $\rho = 0.35, \beta = 0.5$.
- Unsure, but leaning to jump: $\rho = 0.65, \beta = 0.5$.
- Jump is very likely: $\rho = 0.9, \beta = 1$.

Note that since there is a symmetry in the market, the results of the above market also covers the negative variant. We also note that in the simulated markets, both interest rates will have a monotonous trend by design. The is an intuitive limitation as when interest rates are going in one direction, it is very counter-intuitive to already expect jumps in the opposite direction to occur. Since we simulate 8 meetings in our markets and there are 4 possible $(\rho, \beta)$-pairs per meeting, our simulation covers $4^8 \approx 65\ 000$ markets, which means that with the symmetry argument, the simulation provides insight into roughly 130 000 different markets.

The left plot in Figure 7.3 shows on the x-axis the duration allocation risk divided by the benchmark risk

$$\frac{\sigma_{da}}{\sigma_{bm}}$$

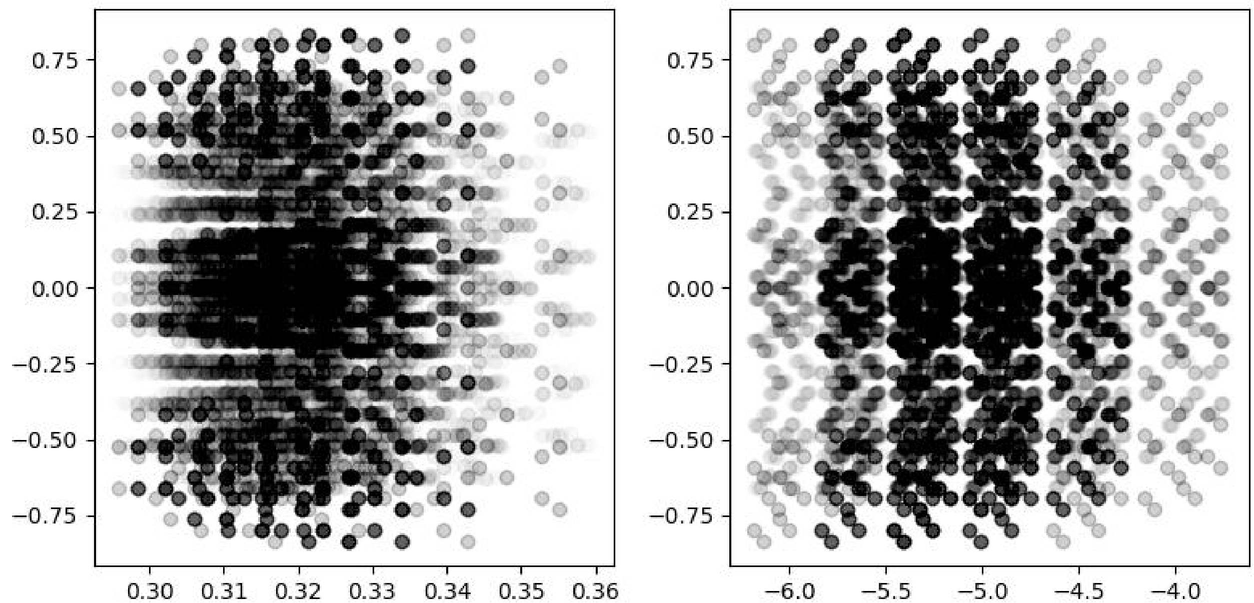


Figure 7.3: Scatter plot of the risk and reward gain of the Duration Allocation Problem vs. the 3-month benchmark

and on the y-axis the difference between the duration allocation profit and the benchmark profit

$$\mu_{da} - \mu_{bm},$$

where the latter is expressed in basepoints of the full invested capital (in the case of our example, €500 million). The right image has the same y-axis, but this time the x-axis shows the risk difference

$$\sigma_{da} - \sigma_{bm},$$

expressed in basepoints of the full invested capital. See that the plots are symmetric on the y-axis, so the duration allocation optimum under performs compared to the benchmark just as often as it outperforms the benchmark when looking at the reward. The risk component tells a very different story. The for this specific structuring of meetings and time horizon, there is a 64-70% reduction in risk using the duration allocation problem compared to the benchmark. We also see that in absolute terms, this is between a 4 and 6 basepoint difference, even if the reward difference is no more than a basepoint. This shows that under the assumptions of the simulation, the duration allocation method is significantly better at reducing risk than the 3-month benchmark.

### 7.2.2 Simulation Parameters

Now that we have run a major simulation to compare the duration allocation result to the 3-month benchmark, we will perform a number of smaller simulations to show the effects of changing the simulation parameters on the results. The first thing we will do is reduce the amount of possible market expectations to 3:

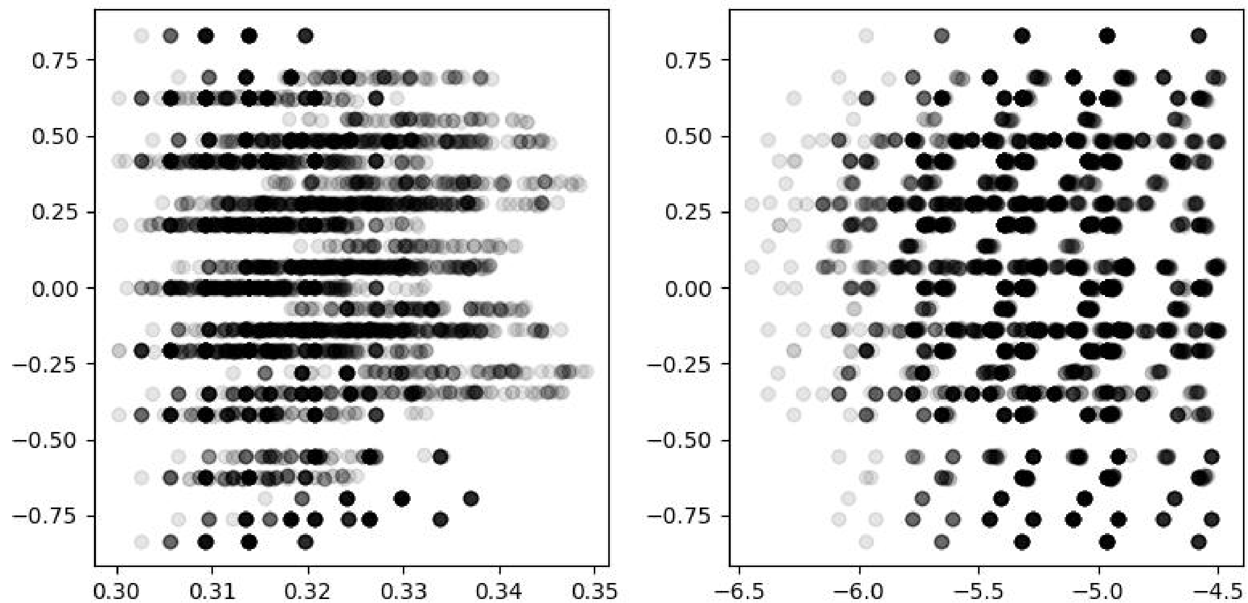


Figure 7.4: A recreation of Figure 7.3 with only 3 market expectation options instead of 4.

- Jump is unlikely: $\rho = 0.15, \beta = 0$.
- Jump is unsure: $\rho = 0.55, \beta = 0.5$.
- Jump is likely: $\rho = 0.85, \beta = 1$.

This reduces the amount of simulations to $3^8 \approx 6500$, a factor 10 reduction. In Figure 7.4, we see that though there is now a slight asymmetry in the market due to the offsetting of $\rho = 0.55, \beta = 0.5$, the result remains largely unchanged. We now proceed to remove the last two time periods, reducing the time horizon by 40 days. This results in just 729 market simulations.

We see the result of this reduction in Figure 7.5. We now see that although the general structure of the plot looks quite similar, the risk reduction is smaller. From 65-70% reduction, we now only have a 50-55% reduction. This is because the removal of the last two time periods means that the duration allocation problem as 2 fewer degrees of freedom. Before the removal, the duration allocation problem had the option of fixing interest rates for 175 days, while now the maximum number of interest days that can be fixed is 135, which means there is more interest rate risk due to FX swaps expiring earlier.

This result can be intuitively generalised to the risk reduction being proportional to the benchmark tenor and the time horizon. If the duration allocation problem is allowed to purchase much longer FX swaps, it is capable of reducing significantly more interest rate exposure.

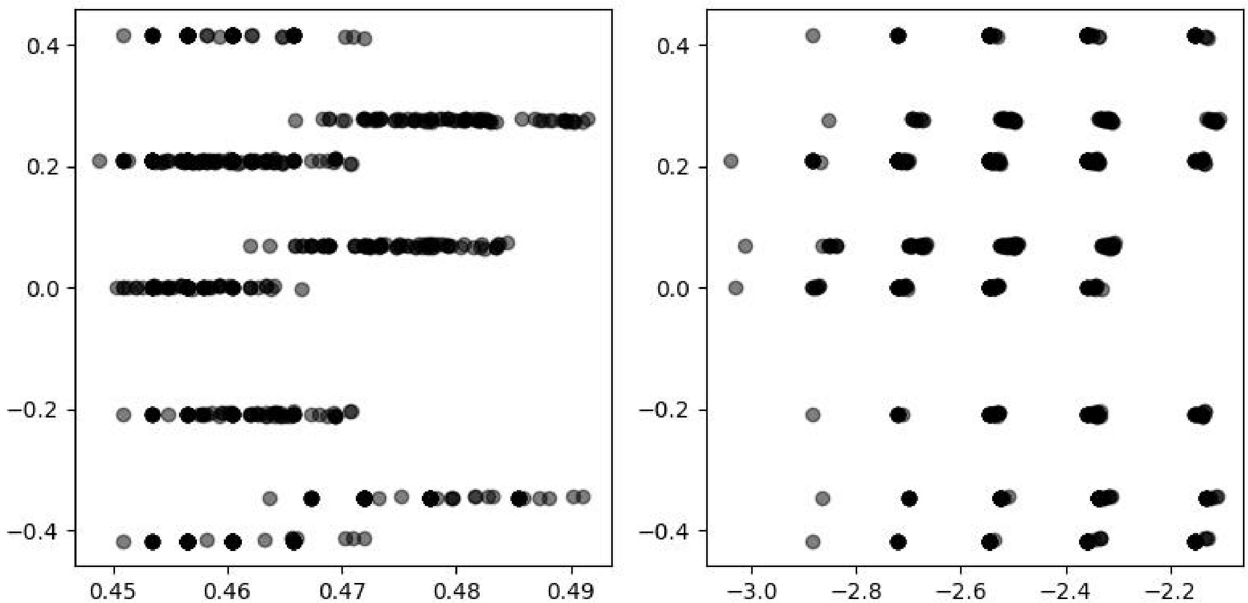


Figure 7.5: A recreation of Figure 7.4 after removing the last two periods.

### 7.2.3 Benchmark Comparison

In order to build upon the claim made in the last subsection, we now proceed to compare the duration allocation results to some additional benchmarks. We still use the 3 options for interest rate expectations to reduce the number of simulations, but this time we introduce the additional 5-month benchmark.[3] We now do again consider the original time partition where we do not remove the last two periods. The results of the new benchmark comparison can be found in Figure 7.6. We notice that the risk reduction compared to this benchmark is smaller, as expected, but we see that there is a less chaotic reward pattern.

From this we conclude that longer tenors provide a stronger risk-reduction, but they may also expose the portfolio to a more clear reward loss (or gain). We should note here that the 5-month benchmark does violate concentration limits and may thus not be usable in practice. We see that despite the 5-months benchmark's ability to fix many interest rate days without a concentration limit, the duration allocation optimum still yields a lower risk by 23-25%.

## 7.3 Outcome Analysis

Now that we have established that the duration allocation optimum outperforms the benchmark in many markets, we return to our example to discuss how the payoff of an FX swap portfolio can be assessed. The outcome of the FX swap portfolio is a function of the realised interest rates, which are still random quantities

[3]We do not use the more intuitive 6-month benchmark, as this would fix too many interest rate days in the given time partition.

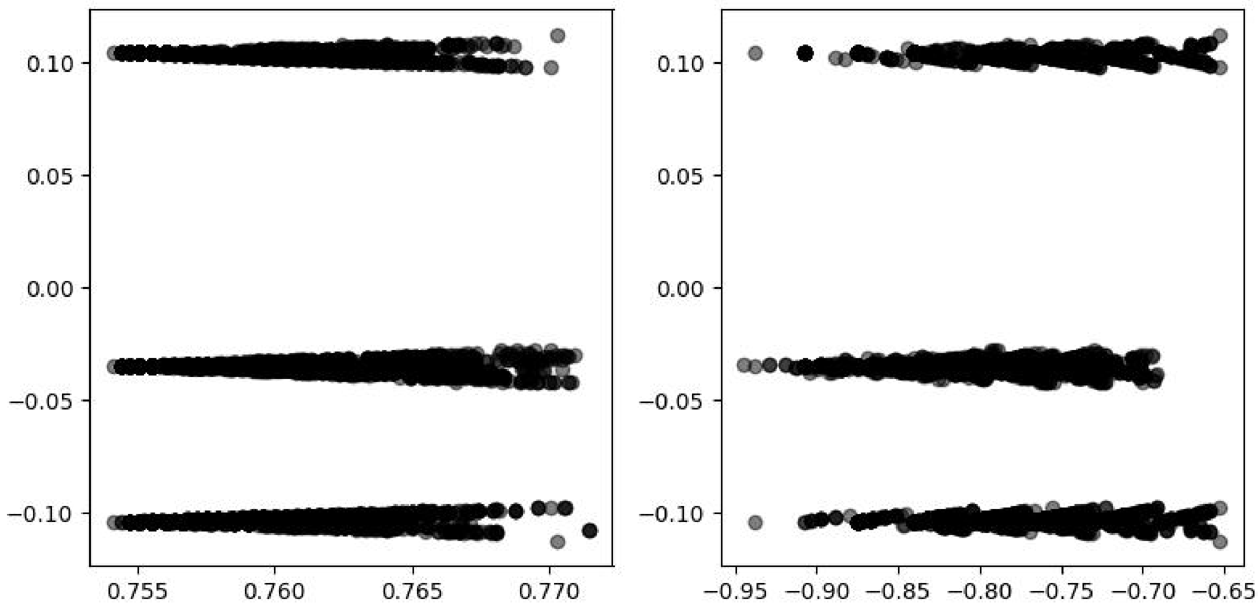


Figure 7.6: A recreation of Figure 7.4 using the 5-month tenor instead of the 3-month tenor as a benchmark

at the time that the swap portfolio is constructed. Since the duration allocation problem of our example only covers 10 meetings, and we assume that any meeting can only result in a jump or no jump, there are $2^{10} = 1024$ different possible outcomes for the payoff of the FX swap portfolio. This is a small enough quantity that we can analyse the risks of the payoffs, especially since some of these possible payouts are very unlikely.

A histogram showing the various FX swap payoffs can be seen in Figure 7.7. We see here that the difference between the lowest and the highest possible payoffs is 0.25% for the optimal FX swap portfolio calculated by the duration allocation problem and 0.5% for the benchmark. We see that the benchmark has a much wider range of possible payoffs, which means there is more risk involved. The shown results disregard the likelihoods of potential outcomes. In order to take these likelihoods into account, we proceed to discard outcomes if they are deemed implausible. We will formulate some criteria based on intuition, but additional criteria can easily be implemented if necessary. We use the following criteria, based on the market data.

- Two consecutive meetings will not both result in jumps: Looking at Table 7.1, we see that the market finds it very unlikely for two consecutive meetings to result in a jump.
- Jump limits: When looking at Table 7.1, we see that there are more than 3 jumps priced in for the euro interest rate and no more than 2 jumps priced in for the dollar interest rate, so we remove the possibility of more jumps occurring.

The histogram resulting from these restriction can be found in Figure 7.8. We summarise a number of statistics in Table 7.7, namely the lower bound, upper bound, sample mean and standard error. Note that for the sample mean and standard error we still do not take the likelihoods of the payoff into account (other than removing very unlikely payoffs in the restricted data). The reason for this is that in order to determine

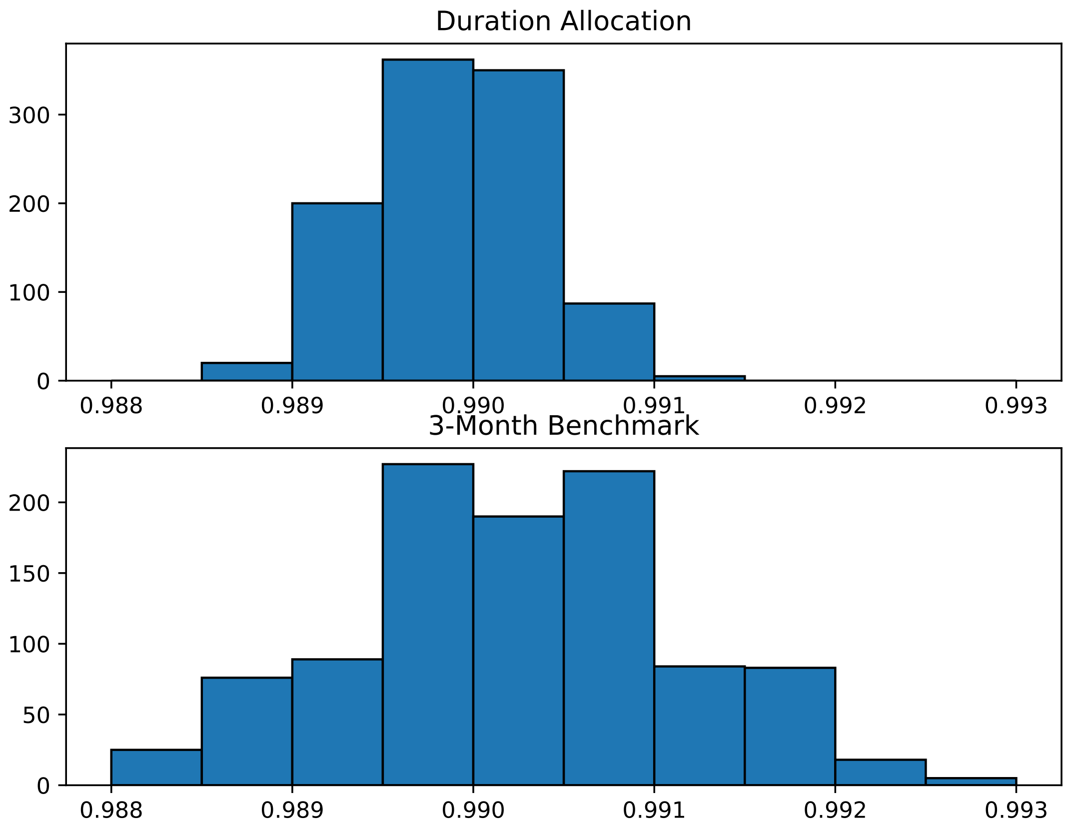


Figure 7.7: Outcome Histogram of the Optimal FX Swap Portfolio

the probabilities of the payoffs, the joint distribution of all the jumps needs to be known, which means the correlation structure between all jumps needs to be known. Although we could impose a correlation structure just like we did when constructing $\Sigma$, we would need to make assumptions which may not necessarily be true in practice. When constructing $\Sigma$, these assumptions were necessary to produce a usable model, but in this case, there is also value in simply considering all plausible outcomes.

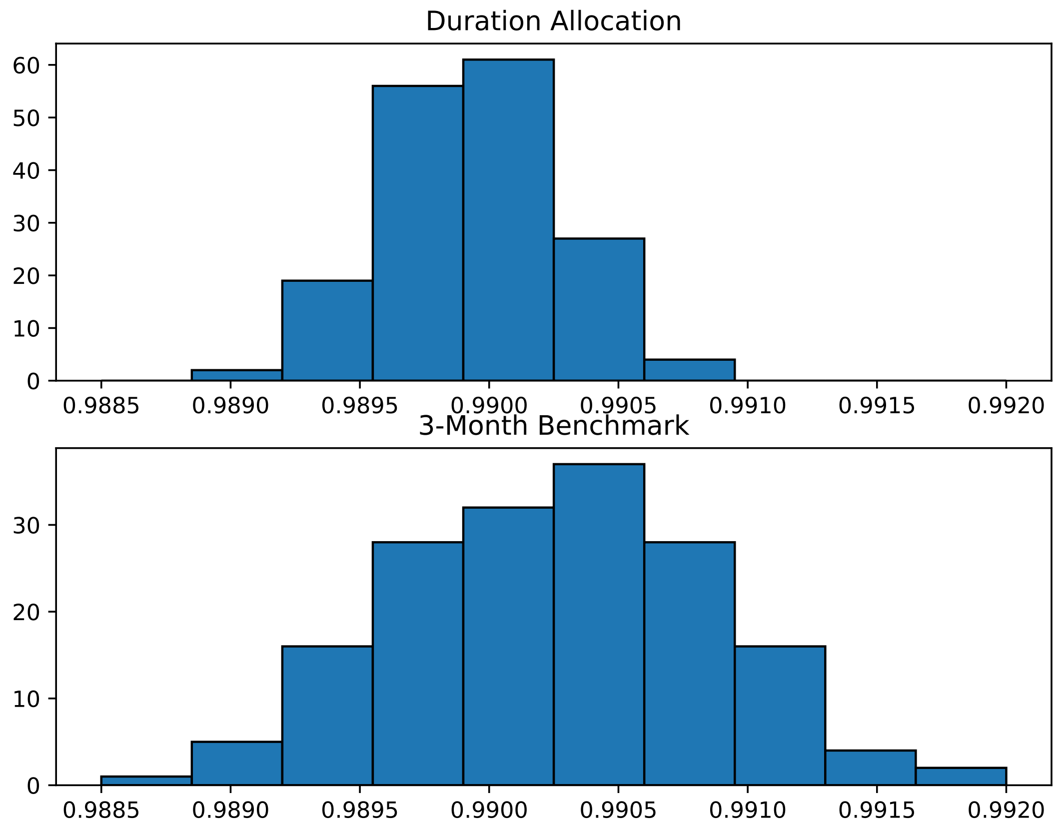


Figure 7.8: Outcome Histogram of the Optimal FX Swap Portfolio with Certain Jump Restrictions

| | Duration Allocation | | 3-Month Benchmark | |
|---|---|---|---|---|
| | No Restriction | Restriction | Restriction | No Restriction |
| Lower Bound | 98.85% | 98.91% | 98.88% | 98.75% |
| Upper Bound | 99.13% | 99.07% | 99.18% | 99.29% |
| Range | 28bps | 16bps | 30bps | 54bps |
| Sample Mean | 98.99% | 98.99% | 99.03% | 99.03% |
| Standard Error | 4.369bps | 2.971bps | 5.986bps | 9.179bps |

Table 7.7: Various Statistics for the Sample Payoffs of the Duration Allocation Optimum and the 3-Month Benchmark

We see in the table that the duration allocation optimum slightly under performs compared to the 3-month benchmark in terms of payoff, but the risk of the benchmark is halved. Furthermore, the range of the outcome is vastly reduced using the duration allocation optimum. We note here that even though the return and risk statistics seem very small, since it is not unrealistic to trade €500 million per day, a difference of 1bps correspond to €50 000 per day. We also reiterate that the main advantage of the above table and histograms is that it can be computed on the day that the swaps are bought, immediately providing a trader with a risk profile.

# Chapter 8

# Conclusion

In this chapter, we summarise the concepts discussed and results obtained in this thesis and formulate a conclusion for each major results. We start by discussing duration and duration equivalence and then continue to the proposed interest rate models. We then discuss the duration allocation problem and the obtained results. We conclude by providing a number of topics for future research.

## 8.1 Duration and Duration Equivalence

We started out showing that FX swaps can be mathematically modelled as sovereign bonds, only depending on central bank interest rates.[1] We then defined the concept of swaps for sovereign bonds and by extension for FX swaps. This new definition mirrors the existing concept of duration defined by Macauley for coupon bonds in that it provides the sensitivity of sovereign bonds and FX swap (portfolios) to changes in interest rates. This concept can be used to provide a simplified representation of the sensitivities of an FX swap of sovereign bond portfolio.

After defining the concept of duration and proving some basic properties, we proved the duration equivalence theorem which states that sovereign bond and FX swap portfolios with the same durations have roughly the same payoff. In the same chapter, we also found a bound on the approximation, which turns out to be very small in practice. We also provided some applications of duration equivalence, namely hedging and portfolio optimization. We also provided a method to find a bond investment with a given duration distribution called the 'vertical bond investment'. The aim of the rest of the thesis was to apply duration equivalence in the context of portfolio optimization.

## 8.2 Interest Rate Modelling

Before moving on to portfolio optimization, we first needed to find a way to convert available market information into usable parameters for an optimization problem. To do this, we first found a way to model the profit resulting from a portfolio optimization given some market information and an estimator for the interest rates.

[1] We reiterate here that modelling bonds this way can only be done mathematically, bonds and swaps are fundamentally different products, but they have the same payout structure.

Modelling the risk component of the optimization problem required more mathematics and financial intuition. We formulated various interest rate models imposing independent, Markovian and manually defined correlation structures. We then derived some properties of each of these interest models and used them to construct a covariance matrix to be used in the portfolio optimization problem.

## 8.3 Duration Allocation

After formulating a method for translating market data and interest rate estimates into optimization parameters, we proceeded to place the FX swap portfolio optimization problem into a well-known optimization framework, for which we chose the Markowitz framework. The duration framework allows us to simplify the FX Swap portfolio optimization problem into the duration allocation problem and we proved a theorem to show that the duration allocation problem is equivalent to the long-only Markowitz problem, which can be solved with numerical solvers. We also proved a handful of theorems surrounding the solution set of the duration allocation problem. We concluded by briefly expanding the model with concentration limits.

We showed how the duration allocation problem can be used in practice and how to translate any optimal solution into instructions for traders. We did this based on a real-world example, where we were able to reduce the risk with 60% compared the standard 3-month benchmark. We constructed a simulator that simulated 65 000 financial markets to try to account for many different market conditions. From these simulations, we concluded that the duration allocation problem outperforms the 3-month benchmark by 65-70% for the risk component without significant losses in return. We concluded by showing some additional properties that can be derived on the day that the swaps are purchased in order to give the trader a better insight into the risk and reward structure of the optimal portfolio.

## 8.4 Further Research

Since the concept of duration for FX swaps and sovereign bonds is entirely new, there is much potential for future research. We list a number of such topics.

- **Short Positions and Futures:**
  The current duration framework is built on bond investments with only positive components, which means that bonds can only be bought, not sold. By also allowing negative components, the portfolios can also include short positions and future contracts. It would be interesting to see if the theorems in this thesis still hold under this generalisation and what effect it would have on the portfolio optimization problem, as well as hedging strategies. Note that one of the key properties of duration distributions no longer holds in this extended framework.

- **Computational Results of Duration Equivalence**
  In this thesis we provided the duration equivalence theorem and an upper bound for the error in this theorem. We saw that the upper bound was sufficiently small in practice to consider the error negligible and verified this with an empirical result. For more general use cases, this error could be significant (think about large positional portfolios) and so it could be interesting to zoom in on the error in computational example. To this end, it would be useful to define some special classes of bond investments and see if general properties can be derived for the entire class.

- **Bond Investment Constructions from Duration Distribution**
  In this thesis, we provided a single method for translating a duration distribution into a bond investment in the form of the vertical bond investment. For out purposes, this single method was sufficient, but many more methods could be formulated for various other applications.

- **Covariance Matrix:**
  This thesis provided a method for modelling the covariance matrix in the duration allocation problem, but in the interest of time, this discussion was brief. It may be interesting to derive properties of the covariance matrix and try to add other components to the covariance matrix such as concentration risk.

- **Additional Properties of the Duration Allocation Problem**
  In this thesis, we proved a number of theorem regarding the duration allocation problem. We mainly proved simple theorems that were directly applicable to our problem, but many other theorems may be formulated. In this regard additional theorems surrounding concentration limits could be interesting results. In addition, this thesis did not dive deep into the existing Markowitz theory, so additional theorems surrounding this theory could be translated to the duration allocation problem.

- **Cross-Currency Basis**
  At the beginning of this report, we stated that we would not take the cross-currency basis into account in order to focus on a better understanding of the duration model. In further research, one could look into the effects of the cross-currency basis on the results of the duration allocation model.

# Bibliography


[1] Interest rate parity (irp) definition, formula, and example, May 2024.

[2] Official interest rates, April 2024.

[3] Taofeek Olusola Ayinde, Abiodun S Bankole, and Oluwatosin Adeniyi. Modelling central bank behaviour in nigeria: A markov-switching approach. *Central Bank Review*, 20(4):213–221, 2020.

[4] European Central Bank. What is the deposit facility rate?, Mar 2016.

[5] European Central Bank. Meetings of the governing council and the general council, Apr 2024.

[6] Pierre Brugiere. *Quantitative portfolio management: With applications in python.* Springer Nature, 2021.

[7] James Chen. Macaulay duration: Definition, formula, example, and how it works, Sep 2022.

[8] James Clear. All models are wrong, some are useful, 2018.

[9] MN. Asset management, Nov 2018.

[10] Troy Segal. Forex market: Who trades currencies and why, Aug 2021.

[11] Frank Smets. Central bank macroeconometric models and the monetary policy transmission mechanism. *Bank for International Settlements*, 3, 1995.

[12] Roman L. Weil. Macaulay's duration: An appreciation. *The Journal of Business*, 46(4):589–592, 1973.

# Appendix A

# Literature Review

In this appendix, we will summarize results from the literature surrounding the topic of FX swap investing. We will divide the literature into 4 categories. The first category includes literature concerned with FX swap and sovereign bond trading and investing strategies. The second part focuses more specifically on existing literature on bond durations. We then briefly discuss literature on interest rate models and conclude with literature on portfolio optimization methods.

## A.1 FX Swap Trading

Since we discussed that we can model FX swaps as bonds in chapter 2, this section contains literature both on FX swap trading/investing and general (sovereign) bond trading/investing.

### A.1.1 Sovereign Bonds

Sovereign bonds are bonds issued by governments and many sovereign bonds have a very small chance to default. This means that the default premium normally obtained when purchasing a bond is negligible, leaving only the risk-free interest rate. When looking at available literature, there are many economic papers analysing qualitative aspects of sovereign bonds, but the only mathematical papers on bonds and other debt securities place a heavy focus on the credit risk of the bond, thus assuming lower credit ratings. This is not entirely unexpected as from a mathematical perspective, sovereign bonds are not very interesting contracts.

Since sovereign bond payouts are simply a deterministic function of the risk-free rate, there is not much modelling necessary to completely characterise a bond if the risk-free rate is known, which is often the case. The only situation where this characterisation gains a stochastic component is when the risk-free rate is predicted to change, which coincides with an ECB meeting. We will discuss this is more detail in section A.3, since this then becomes an interest rate modelling problem.

### A.1.2 FX Swaps

For FX swaps, we see a similar pattern to sovereign bonds as the majority of the literature is more economic and qualitative in nature. FX swaps are not very interesting financial products for the same reason as sovereign bonds; they are entirely dependent on risk-free rates. Since we can not find good models for FX swap trading, we therefore decide to construct our own model from the ground up. We base the model loosely on the exposure of bond payouts to changes in the interest rates. This concept is known in the financial literature as "Duration".

## A.2 Existing Duration Frameworks

In the world of bonds, there is already a concept called 'duration' and it corresponds to "the weighted average term to maturity of the cash flows from a bond" [7]. The formula for Macauley duration is

$$\text{Dur} = \frac{\sum_{t=1}^{n} \frac{tC}{(1+y)^t} + \frac{nM}{(1+y)^n}}{\text{Current Bond Price}}.$$

The $C$ in this formula corresponds to coupon payments and since an FX swap does not have a coupon payment, the formula simplifies to

$$\text{Dur} = \frac{\frac{nM}{(1+y)^n}}{\text{Current Bond Price}}.$$

Here $n$ corresponds to the number of coupon payments, which is 1 since we can see the maturity settlement as a payment. Furthermore, $M$ is the maturity value of the bond and $y$ is the periodic yield. This means that the duration is equal to.

$$\text{Dur} = \frac{\frac{M}{1+y}}{\text{Current Bond Price}} = 1,$$

since $(1+y)$ now corresponds to the yield to maturity. We clearly see that applying Macauley duration on FX swaps (and also zero-coupon bonds) is not very interesting, so we can take the liberty of redefining this concept.

## A.3 Interest Rate Models

In chapter 5, we provide a model for central bank interest rates. Much like sovereign bonds and FX swaps, many of the papers surrounding these interest rates aim to model them using macroeconomics, rather than quantitative methods ([11] is an example). There is a paper that, like this thesis, attempts to use a Markovian model for the interest rates [3], but the dynamics of this model are too complicated to derive the properties that we desire to construct a covariance matrix. Furthermore, the interest model we use needs to be intuitive as we need our models to remain fully explainable, as per the societal obligations of MN.

Here too, we provide some reason as to why there may not be that many quantitative papers. In the past 25 years, there have only been 59 interest rate changes by the ECB [2]. This means that any data-driven model for these interest rates does not have enough training data, and any theoretical model does not have enough validation data. Furthermore, in these past 25 years, the economic landscape has changed drastically though major events such as 9/11, 2008 and Covid, so the assumption that the 59 available data points can be seen as independent and identically distributed is not realistic. Because of this, it is much more practical to create intuitive models for the interest rates based on whichever application the model will be used for, as has been done in this thesis.

## A.4 Portfolio Optimization

As the last step of the thesis, we applied the duration equivalence to construct and solve a portfolio optimization problem. The concept of duration allows us to reshape the problem into a Markowitz problem [6]. Since the proposed duration model is entirely new, we only apply the regular Markowitz problem and do not dive into any extensions, as this would extend this thesis beyond a reasonable length and it would not fit in the main goal of this thesis.

# Appendix B

# Currency Exposure

In this appendix, we briefly cover the motivation behind purchasing FX swap as a tool to hedge currency exposures. When an investor is stationed in one country, but wants to invest in assets in another country which uses another currency, the investor will be exposed to both the asset and the foreign currency. If the asset grows in value, but the currency decreases in value, the investor could still incur a loss. It is natural for an investor to want to cover their currency exposure so that they only have to worry about the risk of the asset.

## B.1 Currency Exposures in Foreign Asset Investing

We assume the investor to have a cash pool in NAT, which they want to invest in assets that are valued in FOR. We denote the asset price (in FOR) at time $t$ by $S_t$ and the conversion rate from NAT to FOR at time $t$ by $C_t$. This means that

$$X \text{ NAT} = C_t \times X \text{ FOR}.$$

Suppose the investor want to invest $X$ NAT into the an asset $S$ with current asset price $S_t$. This allows the investor to purchase

$$\frac{C_t \times X}{S_t}$$

of the asset. at time $T$, the investment will thus be worth

$$\left(\frac{C_t \times X}{S_t}\right)\left(\frac{S_T}{C_T}\right) \text{ NAT} = \left(\frac{S_T}{S_t}\right)\left(\frac{C_t}{C_T}\right) X \text{ NAT}. \tag{B.1}$$

We now see that the investor's return on investment is

$$\left(\frac{S_T}{S_t}\right)\left(\frac{C_t}{C_T}\right)$$

Which will be larger if $S_T >> S_t$ and $C_t >> C_T$, i.e. if the asset price goes up and the exchange rate goes down. This is consistent with our expectation as the investment in the foreign currency will be profitable if both the asset and the foreign currency are doing well.

## B.2 Currency Swaps and Currency Tables

We now seek to remove (or at least reduce) the currency exposure. The method we use utilises the ability for certain swaps to 'fix' the current exchange rate. We still purchase the same amount of the underlying asset, but this time we also enter into a currency swap agreement. The currency swap agreement swaps the base currency with the foreign currency both in the present and the future. We use Table B.1, henceforth referred to as a currency table, to show all relevant payouts for the currency swap.

| Time | $T_0$ | | $T_1$ | |
|---|---|---|---|---|
| Currency | NAT | FOR | NAT | FOR |
| Swap | $-X$ | $+C_t \times X$ | $+X$ | $-C_t \times X$ |
| Total | $-X$ | $+C_t \times X$ | $+X$ | $-C_t \times X$ |

Table B.1: Currency table for a currency swap agreement. $T_0$ is the settlement date and $T_1$ is the maturity date.

We can now use the currency table to verify Equation B.1. This verification is shown in Table B.2

| Time | $T_0$ | | $T_1$ | |
|---|---|---|---|---|
| Currency | NAT | FOR | NAT | FOR |
| Convert NAT to FOR | $-X$ | $+C_t \times X$ | 0 | 0 |
| Buy $\frac{C_t \times X}{S_t}$ stocks | 0 | $-C_t \times X$ | 0 | $\frac{C_t \times X}{S_t} S_T$ |
| Convert FOR to NAT | 0 | 0 | $+\frac{C_t \times X \times S_T}{S_t} \times \frac{1}{C_T}$ | $-\frac{C_t \times X \times S_T}{S_t}$ |
| Total | $-X$ | 0 | $\frac{C_t \times X \times S_T}{S_t \times C_T}$ | 0 |

Table B.2: Currency table for a foreign investment. At both the settlement and maturity date, the capital is converted to and from NAT.

We note that we neglect factors such as interest rates, transaction costs and the price of the swap in order not to over-complicate the base model.

## B.3 Covering the Currency Exposure

Using a currency table, we can see what happens when we combine the foreign asset investment with a currency swap both with the same settlement and maturity date. For the sake of simplicity, we assume that the swap volume is equal to the invested capital on the settlement date. The resulting currency table, Table B.3 shows that at maturity, there is still some of the foreign currency remaining.

We need to convert this remaining currency with the conversion rate at maturity.

$$\left(\frac{S_T}{S_t} - 1\right) C_t \times X \text{ FOR} \qquad \rightarrow \qquad \left(\frac{S_T}{S_t} - 1\right) \frac{C_t}{C_T} \times X \text{ NAT}$$

This results in the capital value at maturity in NAT being

| Time | $T_0$ | | $T_1$ | |
|---|---|---|---|---|
| Currency | NAT | FOR | NAT | FOR |
| Buy $\frac{C_t \times X}{S_t}$ stocks | 0 | $-C_t \times X$ | 0 | $\frac{C_t \times X}{S_t} S_T$ |
| Swap $X$ NAT to FOR | $-X$ | $+C_t \times X$ | $+X$ | $-C_t \times X$ |
| Total | $-X$ | 0 | $+X$ | $\left(\frac{S_T}{S_t} - 1\right) C_t \times X$ |

Table B.3: Currency table for a currency swap agreement. $T_0$ is the settlement date and $T_1$ is the maturity date.

$$X + \left(\frac{S_T}{S_t} - 1\right) \frac{C_t}{C_T} \times X.$$

We now notice that the $X$ term in this value is constant and thus does not contain any risk. The risky component only results in the conversion of the remaining FOR at maturity. This is consistent with expectation as the only foreign currency risk originates from the (stochastic) foreign currency conversion.

## B.4 Comparing the Covered Portfolio with the Exposed Portfolio

Now that we have formulated a payoff formula for both the regular and the covered investment, we compare their basic probabilistic properties. We first recall the two formulae.

$$\begin{aligned}
\text{Regular Investment Payoff (NAT): } f_R(T) &= \left(\frac{S_T}{S_t}\right) \times \left(\frac{C_t}{C_T}\right) \times X, \\
\text{Covered Investment Payoff (NAT): } f_C(T) &= X + \left(\frac{S_T}{S_t} - 1\right) \times \frac{C_t}{C_T} \times X.
\end{aligned}$$

The only two stochastic quantities are $S_T$ (the asset return) and $C_T$ (the conversion rate at maturity). We assume these two random quantities to be independent. We first compare the expected returns.

$$\begin{aligned}
\mathbb{E}\left[f_R(T)\right] &= \mathbb{E}\left[\frac{S_T}{C_T}\right] \frac{C_t}{S_t} X, \\
\mathbb{E}\left[f_C(T)\right] &= \mathbb{E}\left[\frac{S_T}{S_t} - 1\right] \mathbb{E}\left[\frac{C_t}{C_T}\right] X \\
&= \mathbb{E}\left[\frac{S_T}{C_T}\right] \frac{C_t}{S_t} X.
\end{aligned}$$

We see that the two portfolios yield the same expected return. We now proceed to compare the variances. Variance gives us a measure for the portfolio risk. Since we only want to limit the currency exposure, we will condition the variance to fix the asset return.

$$
\begin{aligned}
\operatorname{Var}\left( f_R(T) \middle| S_T \right) &= S_T^2 \operatorname{Var}\left(\frac{1}{C_T}\right) \frac{C_t^2 X^2}{S_t^2} \\
\operatorname{Var}\left( f_C(T) \middle| S_T \right) &= \operatorname{Var}\left( \left(\frac{S_T}{S_t} - 1\right) \frac{1}{C_T} \middle| S_T \right) C_t^2 X^2 \\
&= \left(\frac{S_T}{S_t} - 1\right)^2 C_t^2 X^2 \operatorname{Var}\left(\frac{1}{C_T}\right)
\end{aligned}
$$

We now investigate the relation between the two variances.We recall that we want decrease the risk using the swap, so we want

$$
\begin{array}{rcr}
\operatorname{Var}\left( f_R(T) \middle| S_T \right) & \geq & \operatorname{Var}\left( f_C(T) \middle| S_T \right) \\
S_T^2 \operatorname{Var}\left(\frac{1}{C_T}\right) \frac{C_t^2 X^2}{S_t^2} & \geq & \left(\frac{S_T}{S_t} - 1\right)^2 C_t^2 X^2 \operatorname{Var}\left(\frac{1}{C_T}\right) \\
\frac{S_T^2}{S_t^2} & \geq & \left(\frac{S_T}{S_t} - 1\right)^2 \\
\frac{S_T^2}{S_t^2} & \geq & \frac{S_T^2}{S_t^2} - 2\frac{S_T}{S_t} + 1 \\
2S_T & \geq & S_t
\end{array}
$$

by positivity of the variance, $C_t^2$, $X^2$, $S_t^2$ and $S_T^2$. In other words, the risk arising from the currency exposure is reduced by the swap covering as long as the asset price at maturity $S_T$ is not less than half of the asset price at settlement, $S_t$.

## B.5 Conclusion

We find that the condition $S_T \geq \frac{1}{2} S_t$ is sufficient for the swap strategy to reduce the risk arising from the currency exposure, under the condition that the currency conversion rate and the asset return are independent. This conclusion is consistent with expectation, as a very small asset return results in the swap over-returning the native currency. This results in a significant short position in the foreign currency, creating a large exposure.

# Appendix C

# Bernoulli Correlation

In this appendix, we explore the effects of imposing a correlation structure on Bernoulli random variables. The goal is to link the correlation between two Bernoulli random variables with their condition probabilities. For this sake, consider 2 Bernoulli random variables $X_1 \sim \text{Ber}(p), X_2 \sim \text{Ber}(q)$. We assume these random variables to correspond to two events of which $X_1$ is the first to occur. The correlation between $X_1$ and $X_2$ is given by

$$\begin{aligned}\rho(X_1, X_2) &= \frac{\text{Cov}(X_1, X_2)}{\sigma_1 \sigma_2} \\ &= \frac{\text{Cov}(X_1, X_2)}{\sqrt{p(1-p)q(1-q)}}\end{aligned}$$

The covariance between the two is given by

$$\begin{aligned}\text{Cov}(X_1, X_2) &= \mathbb{E}[X_1 X_2] - \mathbb{E}[X_1]\mathbb{E}[X_2] \\ &= \mathbb{P}(X_1 = X_2 = 1) - pq \\ &= \mathbb{P}(X_2 = 1 | X_1 = 1)\mathbb{P}(X_1 = 1) - pq \\ &= \mathbb{P}(X_2 = 1 | X_1 = 1)p - pq\end{aligned}$$

If we now set $\mathbb{P}(X_2 = 1 | X_1 = 1) = x$ and simplify the expression for the correlation, we get

$$\begin{aligned}\rho(X_1, X_2) &= \frac{p(x-q)}{\sqrt{pq(1-p)(1-q)}} \\ &= \frac{\sqrt{p}(x-q)}{\sqrt{q(1-p)(1-q)}}.\end{aligned}$$

If we now solve this equation for $x$, we get

$$x = \frac{\rho(X_1, X_2)\sqrt{q(1-p)(1-q)}}{p} + q.$$

The above formulas can be used to convert correlation into conditional probability and vice-versa.

# Appendix D

# Business Chapters

In this appendix, we discuss the implications of the duration framework from a business perspective. We will discuss a number of things that are not necessarily of mathematical interest, but are easily applicable. In this section, we also briefly summarise some of the concepts in a more intuitive way, without providing any mathematical rigor.

## D.1 Duration

The concept of duration can be intuitively explained as the amount of time an FX swap runs in a given period. The periods correspond to the time between two central bank meeting. In order to sensibly define duration, we also need a time horizon. This is the maximum number of days we consider (which we can set to coincide with the longest swap tenor). If we assume the banks to meet every 20 days (both banks individually every 40 days), then we split the time horizon into periods of 20 days. If the time horizon is 60 days, we get 3 periods.

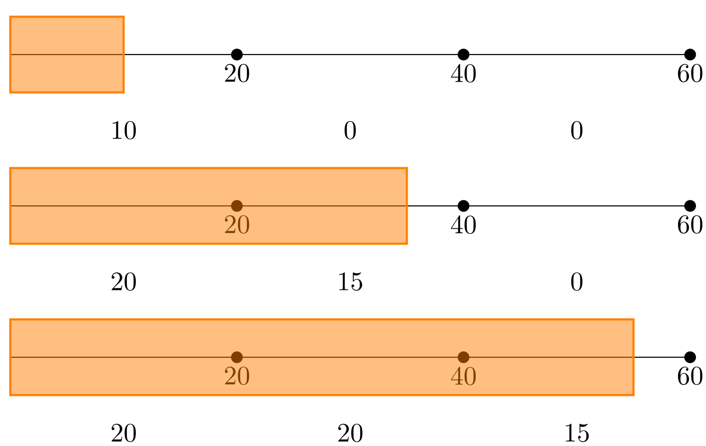


Figure D.1: Duration Distribution for 10-, 35- and 55-day FX swaps over a time horizon of 60 days with a meeting every 20 days.

In Figure D.1, we see the durations of various swaps over the 60-day time horizon. We call the sequence of durations (for example (10, 0, 0) for the 10-day swap) the duration distribution of the swap. In the case that we want to consider a portfolio op swaps instead of individual swaps, we just add the durations together. The duration distribution of the portfolio of the above three swaps is thus $(50, 35, 15)$.

## D.2 FX Swap Payoffs

The duration framework provides us with a simplified method to assess swap portfolio payoffs and exposures. In order to assess these payoffs and exposures, we need the expected values of the interest rate differences for each period. This data can be found in WIRP in Bloomberg. We will refer to the interest rate difference in the $i$'th period by $r_i$ and the WIRP value difference of this period by $\mathbb{E}\left[r_i\right]$.

In Proposition 4, we discussed that if we have an FX swap, we can split its payoff into per-day interest difference components. In the case of the 35-day swap, the payoff will look like[1]

$$\pi_h(S_{35}) = \exp\left(20\mathbb{E}\left[r_1\right] + 15\mathbb{E}\left[r_2\right]\right).$$

Since we purchase these swaps in the market, the interest rate differences we get payed out correspond to what the market prices in, which is the WIRP data. This is why we use the expected values of the interest rate differences. We see that the coefficients in this exponent correspond exactly with the duration distribution of the swap. When the FX swap expires, we need to purchase new swaps with interest rate differences we do not know yet. The values $r_1, r_2$ and $r_3$ correspond to these unknown values, so the payoff at the end of the time horizon will be

$$\pi(S_{35}) = \exp\left(20\mathbb{E}\left[r_1\right] + 15\mathbb{E}\left[r_2\right] + 5r_2 + 20r_3\right).$$

We call the value $\pi_h(S_{35})$ the head of the payoff and the value $\pi_t(S_{35}) = \exp\left(5r_2 + 20r_3\right)$ the tail of the payoff. Note that $\pi(S_{35}) = \pi_h(S_{35})\pi_t(S_{35})$.

## D.3 Portfolio Payoffs

Evaluating the payoff of a portfolio of FX swaps is easily done by simply adding up all FX swap payoffs. The downside of this method is that it is very difficult to determine which exposures the portfolio has. This can be resolved by calculating the payoff using the duration distribution of the portfolio. In Definition 28, we defined a formula called the 'flat payoff' of the portfolio. We also showed here that the flat payoff is approximately equal to the portfolio payoff.

$$\pi_F := L\exp\left(\frac{1}{L}\sum_{k=1}^{N} D_k\mathbb{E}\left[r_k\right] + D_k^C r_k\right)$$

[1] We use exponential interest notation. In regular notation, this would be $(1+r_1)^{20}(1+r_2)^{15}$, which is roughly equal to the provided expression.

The number $L$ is here the number of FX swap bought, $D_k$ is the duration of the portfolio in the $k$'th period and $D_k^C$ is the complement of this duration (in the above 35-day swap, this is $(0, 5, 20)$, the "complement" of $(20, 15, 0)$). Note that the portfolio of this context only consists of FX swaps bought on the same day. This thesis does not provide a method for comparing swaps across multiple days. Further research is required to develop a method for doing this.

By using the flat payoff, it becomes a lot clearer how the FX swap portfolio is exposed to the different interest rate differences. The exposure of a financial product to an underlying value is given by its mathematical derivative, which we often divide by the value of the product to normalise it. This is given by

$$\frac{\frac{d}{d\mathbb{E}[r_k]}\pi_F}{\pi_F} = D_k,$$

so we see that the exposure of the FX swap portfolio to one of the interest rate differences is the duration. If we have a portfolio with duration distribution $(1200, 500, 50, 0)$, a change in the WIRP value of the first period will massively impact the performance of the portfolio, where an increase in the last period will have no impact. If we would like to increase the exposure to the 3rd period, but not the 4th, we should purchase swaps that expire at the end of the 3rd period.

The above analysis only works on the single-day portfolios and does not take into account that we are forced to purchase swaps even if conditions are not favourable.[2] The swap portfolio $(1200, 500, 50, 0)$ has contains a lot of swaps expiring in the near future. These swaps will need to be rolled over into new swaps and if the interest rate difference has since decreased,[3] purchasing new swaps would be less favourable. This means that having a high duration in a given period can be seen as being 'long' that period, but having a low (but still positive) duration in a given period can be seen as being 'short' that duration. In order to define a 'neutral' position in a period, a benchmark would need to be used. Any deviation from the benchmark portfolio's duration would give a long or short position.

The long/short position analysis can be used to analyse multi-day portfolios, as being very long on period 2 today and being very short period 2 tomorrow cancel each other out, as the exposure to to period 2 becomes.

$$\frac{\frac{d}{d\mathbb{E}[r_k]}\pi_F(\text{Today})}{\pi_F(\text{Today})} - \frac{\frac{d}{d\mathbb{E}[r_k]}\pi_F(\text{Tomorrow})}{\pi_F(\text{Tomorrow})} = D_k(\text{Today}) - D_k(\text{Tomorrow}).$$

## D.4 Portfolio Optimization

In addition to comparing FX Swap portfolios, the duration framework can also be used to optimize FX Swap portfolios. We note here that since the duration framework should mainly be used to analyse per-day portfolios, the "Portfolio" we refer to is not the full multi-day FX Swap portfolio, but rather the collection of FX Swaps that should be purchased on any given day.

[2]We can purchase overnight swaps to hope that conditions become more favourable, but this can not be done without limits and comes with significant risks.

[3]The interest rate difference is always the native interest rate minus the foreign interest rate.

The FX Swap portfolio optimization problem is solved by expressing the problem in terms of duration. The resulting problem is called the ”Duration Allocation” problem and is based on the ideas of the Markowitz Optimization problem, which is a frequently used method in classical portfolio theory. Solving the duration allocation problem provides you with a solution in terms of durations, which can be translated to a solution in terms of FX Swap tenors.

The primary inputs of the duration allocation problem are the following:

- **Dates**: The dates of the central bank meetings.
- **Market Expectations of Interest Jumps**: This data can be found under the WIRP ticker in Bloomberg.
- **Interest Jump Estimators**: These correspond to what a estimation model expects to be the result of central bank meetings.
- **Concentration Limits**: These are restrictions on how many FX swaps are allows to expire in between two central bank meetings. These values can differ per inter-meeting period and are expressed as a percentage.

We will discuss the estimator in more detail later on. In addition to these inputs, some additional parameters can be used to modify the model. These are the AMJ-covariance parameter and the DTA-covariance parameter. Default values for these parameters are 1 and 2, see chapter 5 to see how to tune these parameters.[4]

We now dive into how the estimators can be constructed. There are three main ways to do this.

- **Data-Driven Estimators**: If we have access to an estimation method for predicting the results of the central bank meetings, the resulting estimator can be used as an input to the model.
- **Trader-Driven Estimators**: If a trader has a strong inclination for what the result of a central bank meeting will be, this prediction can be used as an input to the model.
- **Simple Estimators**: If there is not good estimate for the future interest rate jumps, we can simply round the market expectations to whole (or half) numbers. This approach results in a portfolio that primarily takes risk into account.

The different estimators will result in different optimal portfolios, so a trader should decide the best course of action as well as how to translate the model’s suggestions to real-world actions.

## D.5 The Portfolio Tool

We now discuss how the duration allocation portfolio tool can be used in practice. The inputs for the model can be plugged into a template in excel, which can be seen in Figure D.2.

[4]Tuning these parameters is not necessary, so only do this if the model is well-understood

| Meeting | Days | Bank | Coef | Jump | Net Jump | Cum Diff | Diff Estimate | Concentration Limit |
|---|---|---|---|---|---|---|---|---|
| | 0 | - | - | - | - | INTEREST DIFF TODAY | - | - |
| | 0 | | 0 | | 0 | 0 | | 30 |
| | 0 | | 0 | | 0 | 0 | | 30 |
| | 0 | | 0 | | 0 | 0 | | 30 |
| | 0 | | 0 | | 0 | 0 | | 30 |
| | 0 | | 0 | | 0 | 0 | | 30 |
| | 0 | | 0 | | 0 | 0 | | 30 |
| | 0 | | 0 | | 0 | 0 | | 30 |
| | 0 | | 0 | | 0 | 0 | | 30 |
| | 0 | | 0 | | 0 | 0 | | 30 |
| | 0 | | 0 | | 0 | 0 | | 30 |

Figure D.2: Input Template for the Duration Allocation Problem

The columns of the template correspond to the following information.

- **Meeting:** The date of the central bank meetings. The first entry should be the current date.
- **Bank:** The Central Bank holding the meeting.
- **Jump:** The probability of a jump during that meeting in percentages multiplied by 10 (500 is 50%).
- **Diff Estimate:** The estimator for the cumulative jumps up to and including that meeting.
- **Concentration Limit:** The percentage of the full invested capital that is allowed to expire in between the previous meeting and the meeting of that row.

An example of an input can be found in Figure D.3. The example input corresponds to the market information of 21st of June, 2024.

| Meeting | Days | Bank | Coef | Jump | Net Jump | Cum Diff | Diff Estimate | Concentration Limit |
|---|---|---|---|---|---|---|---|---|
| 21-May | 0 | - | 0 | 0 | 0 | -1.5 | 0 | - |
| 6-Jun | 16 | ECB | 1 | -957 | -957 | -0.957 | -1 | 30 |
| 12-Jun | 22 | FED | -1 | -26 | 26 | -0.931 | -1 | 30 |
| 18-Jul | 58 | ECB | 1 | -207 | -207 | -1.138 | -1 | 30 |
| 31-Jul | 71 | FED | -1 | -207 | 207 | -0.931 | -1 | 30 |
| 12-Sep | 114 | ECB | 1 | -693 | -693 | -1.624 | -2 | 30 |
| 18-Sep | 120 | FED | -1 | -512 | 512 | -1.112 | -1 | 30 |
| 17-Oct | 149 | ECB | 1 | -276 | -276 | -1.388 | -1 | 30 |
| 7-Nov | 170 | FED | -1 | -339 | 339 | -1.049 | -1 | 30 |
| 12-Dec | 205 | ECB | 1 | -587 | -587 | -1.636 | -2 | 30 |
| 18-Dec | 211 | FED | -1 | -568 | 568 | -1.068 | -1 | 30 |

Figure D.3: Input Example for the Duration Allocation Problem

The information should be stored in the folder 'input' with the title being the current date using the format YYYY-MM-DD. Once the file is in the correct directory and the corresponding `DATE` parameter is changed in the `main.py` file, the `main.py` file can be ran and the result can be found in the 'output' folder. The Python terminal will also provide the Markowitz Bullet, the benchmark risk and return and the minimal-risk portfolio with the corresponding risk and return. The output file contains 100 possible optimal portfolios with their corresponding risk and return in case the minimal-risk portfolio is not viable.

In case the specific parameters of the model need to be changed, they can be found at the top of the `main.py` file. The default values are `JUMP_SIZE = 0.0025` (0.25%), `CAPITAL = 500 000 000` and `AMJ_PARAM = 1, DTA_PARAM = 2`.

# Appendix E

# Code

In this appendix, we show the code used to generate the computations used in this thesis. The code blocks are places in sections and subsections according to their names and directories.

## E.1 market_framework

### E.1.1 interest_model.py

```python
import numpy as np

from portfolio_tool.parameter_construction.covariance_functions import ijm_covariance, simple_var


class TimePartition:
    """This class corresponds to the time partitions. The periods partition the time horizon"""
    def __init__(self, partition: list):
        self.partition = partition
        self.terminal_time = partition[-1]

        self.N = len(self.partition)

    def __getitem__(self, item):
        if item == 0:
            return [0, self.partition[item]]

        return [self.partition[item - 1], self.partition[item]]

    def norm(self):
        """Returns the length of each period in the partition"""
        return [self[i][1] - self[i][0] for i in range(len(self.partition))]


class InterestModel:
    """This class defines the interest rate market we are working in"""
    def __init__(self,
                 time_partition: TimePartition,
                 initial_interest: float = None,
                 jump_size: float = None,
```

```
                 eta_i: list = None,
                 eta_m: list = None):
        """The initial_interest, jump_size, eta_i and eta_m are optional parameters, only necessary when calculating
        the payoff of Bonds/FX Swaps"""
        self.time_partition = time_partition
        self.r0 = initial_interest
        self.jump_size = jump_size
        self.eta_i = eta_i
        self.eta_m = eta_m

        if None in [jump_size, eta_i, eta_m, initial_interest]:
            self.beta = self.rho = None
        else:
            self.beta = np.array([self.r0 + self.jump_size * eta for eta in eta_i])
            self.rho = np.array([self.r0 + self.jump_size * eta for eta in eta_m])

    def variance(self):
        """Return the variance of the market expectation of the interest rates (from WIRP)"""
        return simple_var(self.eta_m)

    def covariance(self, covariance_method, truncation_method):
        """
        :param covariance_method: Markovian Covariance Methods, Manual Covariance or Independent Covariance
        :param truncation_method: Tridiagonal Truncation or Damped Truncation
        :return:
        """
        jump_covariance = covariance_method(self.eta_m)
        aggregate_covariance = truncation_method(self.variance(), jump_covariance)

        return ijm_covariance(aggregate_covariance, self.jump_size)

```

## E.2 duration_framework

### E.2.1 bond_investment.py

```
import numpy as np

from portfolio_tool.market_framework.interest_model import TimePartition, InterestModel


class BondInvestment:
    """This class provides objects that correspond to the Bond Investments, which can also be viewed as
    FX Swap portfolios"""
    def __init__(self, vector: list[float] | np.ndarray, interest_model: InterestModel | TimePartition | list[int]):
        if type(interest_model) is InterestModel:
            self.interest_model = interest_model
            self.time_horizon = interest_model.time_partition

        elif type(interest_model) is TimePartition:
            self.interest_model = None
```

```
            self.time_horizon = interest_model

        else:
            self.interest_model = None
            self.time_horizon = TimePartition(interest_model)
        while len(vector) < self.time_horizon.terminal_time:
            vector.append(0)
        self.vector = vector

    @staticmethod
    def from_dict(dic: dict, interest_model: InterestModel | TimePartition | list[int]) -> 'BondInvestment':
        """Generates a bond investment based on a dictionary using a reduced format"""
        if type(interest_model) is InterestModel:
            terminal_time = interest_model.time_partition.terminal_time
        else:
            terminal_time = interest_model.terminal_time
        vec = np.zeros(terminal_time)
        for key in dic:
            vec[key] = dic[key]
        return BondInvestment(vec, interest_model)

    def capital_limit(self) -> float:
        """Return the capital limit of the bond investment"""
        return sum(self.vector)

    def concentration(self) -> float:
        """Returns the concentration of the bond investment (not currently used)"""
        return np.sqrt(sum([val ** 2 for val in self.vector])) / self.capital_limit()

    def get_terminations(self, average=False) -> list[float]:
        """Return the termination vector of the bond investment. If average is True, the termination values
        are divided by the period lenghts"""
        res = []
        old_b = 0
        for i, b in enumerate(self.time_horizon.partition):
            print(b)
            term = sum(self.vector[old_b: b])
            if average:
                term /= self.time_horizon.norm()[i]
            res.append(term)

            old_b = b
        return res

    def get_durations(self, average=False, relative=False) -> np.ndarray:
        """Returns the duration distribtuion of the bond investment. If average is True, the duration values
        are divided by the period lenghts. If relative is True, the duration values are divided by the first duration
        value"""
        res = []
        norm = self.time_horizon.norm()

        normalizer = None

        for i, b in enumerate([0] + self.time_horizon.partition[:-1]):
```

```
            dur = 0
            for j in range(self.time_horizon.terminal_time - b):
                dur += min(j + 1, norm[i]) * self.vector[b + j]
            if average:
                dur /= norm[i]
            if normalizer is None:
                normalizer = dur
            if relative:
                dur /= normalizer
            res.append(dur)
        return np.array(res)

    def get_effective_durations(self, average=False) -> list[float]:
        """Returns the effective duration vector"""
        res = []
        norm = self.time_horizon.norm()
        for i, b in enumerate([0] + self.time_horizon.partition[:-1]):
            dur = 0
            for j in range(norm[i]):
                dur += (j + 1) * self.vector[b + j]
            if average:
                dur /= norm[i]
            res.append(dur)
        return res


    def get_payoff(self, realized_interest: list[float] | np.ndarray = None):
        """Returns the payoff of the Bond Investment / FX Swap Portfolio. If the interest rate deviates from the
        interest rates in the Interest Rate Model used, then the realized_interest parameter can be used to replace
        the beta"""
        payoff = 0
        if realized_interest is None:
            realized_interest = self.interest_model.beta
        for i in range(self.time_horizon.terminal_time):
            exponent = 0
            old_b = 0
            for k, b in enumerate(self.time_horizon.partition):
                r = realized_interest[k]
                rho = self.interest_model.rho[k]
                for j in range(old_b, b):
                    if j < i:
                        exponent += rho
                    else:
                        exponent += r
                old_b = b
            payoff += self.vector[i] * np.exp(exponent)

        return payoff

    @staticmethod
    def even(values: list, interest_model: InterestModel, separation: list[int] = None):
        """Creates a bond investment with the same amount of expiries each day. This method is not used and
        the separation parameter can be ignored"""
        out = []
        norm = interest_model.time_partition.norm()
```

```python
        for i in range(len(norm)):
            xi = values[i]
            W = 0
            if separation is not None:
                W = separation[i]
            for j in range(norm[i]):
                if j < W:
                    out.append(0)
                else:
                    out.append(xi)
        return BondInvestment(out, interest_model)

    def to_image(self, default: float = -1):
        """Creates a matrix that, when plotted, gives a visual representation of the bond investment
    """
        y_len = len(self.interest_model.time_partition.partition)
        x_len = max(self.interest_model.time_partition.norm())

        part = [0] + self.interest_model.time_partition.partition

        out = []
        for i in range(y_len):
            row = []
            for j in range(x_len):
                if j < self.interest_model.time_partition.norm()[i]:
                    row.append(self.vector[part[i] + j])
                else:
                    row.append(default * np.max(self.vector))
            out.append(row)
        return out

    def __str__(self):
        out = ''
        old_val = 0
        for val in self.time_horizon.partition:
            out += f'{self.vector[old_val:val]}\n'
            old_val = val
        return out

    def simple_representation(self) -> str:
        """Shows a simplified representation of the bond investment"""
        out = ""
        for i in range(len(self.vector)):
            if self.vector[i] > 0.00001:
                out += "{ind:4}: {val: 5}\n".format(ind=i, val=round(self.vector[i],3))
        return out


    @staticmethod
    def spread_benchmark(interest_model: InterestModel, capital=1):
        """Creates a spread benchmark, where the same amount of bonds/swaps expire on every day"""
        partition = interest_model.time_partition
        val = capital / partition.terminal_time
        return BondInvestment.even([val for _ in range(len(partition.partition))], interest_model)

```

### E.2.2 duration_distribution.py

```
import numpy as np
from matplotlib import pyplot as plt

from library.funcs import closest_in_list
from portfolio_tool.duration_framework.bond_investment import BondInvestment
from portfolio_tool.market_framework.interest_model import TimePartition, InterestModel


class DurationDistribution:
    def __init__(self,
                 durations: list | np.ndarray,
                 time_partition: TimePartition,
                 average=False):
        """The average parameter creates the average duration distribution"""
        if average:
            dur = []
            for i in range(len(durations)):
                dur.append(durations[i] * time_partition.norm()[i])
            self.durations = np.array(dur)
        else:
            self.durations = np.array(durations)
        self.time_partition = time_partition

        self.n_period = len(self.durations)

    def __str__(self):
        full_partition = [0] + self.time_partition.partition
        out = ""
        for i in range(self.n_period):
            c1 = f'[{full_partition[i]}, {full_partition[i+1]}]'
            c2 = self.durations[i]
            c3 = self.average()[i]
            out += '{c1:12}: {c2:5}, {c3:5}\n'.format(c1=c1, c2=c2, c3=round(c3,2))
        return out

    def normalized(self):
        """Returns the duration distribution normalised on the first period's duration"""
        return DurationDistribution(self.durations / self.durations[0], self.time_partition)

    def relative(self):
        """Returns the normalized duration distribution vector"""
        return [dur / self.durations[0] for dur in self.durations]

    def average(self):
        """Returns the average duration distribution vector"""
        return [self.durations[i] / self.time_partition.norm()[i] for i in range(len(self.durations)
    )]

    def diff(self):
        """Returns a list of the duration difference between consecutive periods. Corresponds to the
     number of
        swaps that should expire at the end of each period"""
        return [round(self.average()[i] - self.average()[i+1] ,2) for i in range(len(self.durations)
    -1)]

    def vertical_bond_investment(self, interest_model: InterestModel, capital=None):
```

```python
        """Creates the vertical bond investment based on the duration distribution"""
        time_horizon = interest_model.time_partition
        vector = np.zeros(time_horizon.terminal_time)
        future_expiries = 0
        for i in range(len(self.durations) - 1):
            a = time_horizon.partition[-i - 2]
            b = time_horizon.partition[-i - 1]

            effective_duration = self.durations[-i - 1] - future_expiries * (b - a)

            expiry = effective_duration / (b - a)
            vector[b - 1] = expiry
            future_expiries += expiry

        a = 0
        b = time_horizon.partition[0]
        effective_duration = self.durations[0] - future_expiries * (b - a)

        vector[b - 1] = effective_duration / (b - a)

        if capital is not None:
            vector *= capital / np.sum(vector)

        return BondInvestment(vector, interest_model)

    def flat_payoff(self, irp, eip, capital=None):
        """Calculates the flat payoff of the duration distribution"""
        if capital is None:
            capital = self.average()[0]
        periods = self.time_partition.partition
        periods_ = [0] + periods
        period_lengths = [periods_[i + 1] - periods_[i] for i in range(len(periods))]
        exp_lst = [self.durations[i] * eip[i] + (period_lengths[i] * capital - self.durations[i]) * irp[i] for i in
                   range(len(periods))]
        return capital * np.exp(sum(exp_lst) / capital)

    def flat_remainder_term(self, interest_model: InterestModel, full: BondInvestment = None):
        """Calculates the difference between the flat payoff and the payoff of the vertical bond investment"""
        if full is None:
            full = self.vertical_bond_investment(interest_model)
        payoff = full.get_payoff()
        flat_payoff = self.flat_payoff(interest_model.beta, interest_model.rho)

        return payoff - flat_payoff

    def bond_investments(self, interest_model: InterestModel, capital):
        """Creates the vertical bond investments, as well as the spread benchmark and the 3-month benchmark"""
        vertical = self.vertical_bond_investment(interest_model, capital)
        benchmark_spread = BondInvestment.spread_benchmark(interest_model, capital)

        m3 = closest_in_list(180, interest_model.time_partition.partition)
        benchmark_3m = BondInvestment.from_dict({m3: capital}, interest_model)

        return benchmark_spread, benchmark_3m, vertical

```

```python


    def payoff_list(self,
                    interest_model: InterestModel,
                    realised_interest: list[float],
                    capital: float = 1):
        """Returns the flat payoff and the payoffs of the spread benchmark, 3M benchmark and vertical bond investment"""

        bm_spread, bm_3m, vertical = self.bond_investments(interest_model, capital)

        payoffs = [self.flat_payoff(realised_interest, interest_model.rho)*vertical.capital_limit(),
                   bm_spread.get_payoff(realised_interest),
                   bm_3m.get_payoff(realised_interest),
                   vertical.get_payoff(realised_interest)]

        return payoffs

    def lambda_payoff_list(self, interest_model: InterestModel, lam: float, capital: float = 1):
        """DEPRECATED. Use interest estimator accuracy to determine expected payoff"""
        realised_interest = interest_model.rho + lam * (interest_model.beta - interest_model.rho)
        return self.payoff_list(interest_model, realised_interest, capital)

    def correctness_curves(self,
                           interest_model: InterestModel,
                           capital: float = 1,
                           lam_range: list = None,
                           N: int = 100):
        """DEPRECATED. Create curves for expected payoff in function of interest estimator accuracy
"""
        if lam_range is None:
            lam_range = [-1 / 2, 3 / 2]

        curves = []
        lam_lst = []

        for i in range(N + 1):
            lam = lam_range[0] + i * (lam_range[1] - lam_range[0]) / N
            payoffs = self.lambda_payoff_list(interest_model, lam, capital)

            curves.append(payoffs)
            lam_lst.append(lam)

        curves = np.array(curves).T

        return lam_lst, curves

    def plot_correctness_curves(self,
                                interest_model: InterestModel,
                                capital: float = 1,
                                lam_range: list = None,
                                N: int = 100,
                                plot = False):
        """DEPRECATED. Plot curves for expected payoff in function of interest estimator accuracy"""
        lam_lst, curves = self.correctness_curves(interest_model, capital, lam_range, N)
        legend = ["Flat", "Benchmark", 'Full']
        if plot:
            plt.figure(figsize=(10, 6))
```

```
        for i, curve in enumerate(curves):
            plt.plot(lam_lst, curve, label=legend[i])
        plt.legend()
        if plot:
            plt.show()
```

## E.3 parameter_construction

### E.3.1 covariance_functions.py

```
import numpy as np


def jumps_from_eta(eta: list):
    """Provided a cumulative jump list, this function returns the jump sizes"""
    eta_ = [0] + eta
    return [eta_[i + 1] - eta_[i] for i in range(len(eta))]

def simple_var(eta):
    """This function determines the variance vector of a cumulative jump list"""
    jumps = jumps_from_eta(eta)
    return [abs(jump) - jump ** 2 for jump in jumps]


def dmj_covariance(eta):
    """This function calculates the markovian covariance matrix of a cumulative jump list
    based on the 'directional Markov Jump' model"""
    jumps = jumps_from_eta(eta)
    out = []
    for i in range(len(eta) - 1):
        if jumps[i] + jumps[i + 1] >= 1:
            out.append(-(1 - jumps[i]) * (1 - jumps[i + 1]))
        else:
            out.append(-jumps[i] * jumps[i + 1])
    return out


def amj_covariance(eta: list, phi: float = 1):
    """This function calculates the markovian covariance matrix of a cumulative jump list
    based on the 'alternating Markov Jump' model. The parameter phi expresses how much more likely a
     down-jump
    is after an up-jump"""
    jumps = jumps_from_eta(eta)
    out = []
    for i in range(len(eta)-1):
        out.append(jumps[i] * jumps[i+1] * (phi/(1+(phi-1)*jumps[i])-1))
    return out


def tridiag_covariance(var, covar):
    """Provided a variance and markovian covariance vector, this function returns the aggregated
     covariance matrix
    in the form of a tridiagonal matrix"""
    dim = len(var)
    out = np.zeros((dim, dim))
```

```
    for i in range(dim):
        out[i,i] = var[i]
    for i in range(dim-1):
        out[i+1, i] = covar[i]
        out[i, i+1] = covar[i]
    return out


def dta_covariance(var_vector: list, cov_vector: list, damping: int = 2):
    """Provided a variance and markovian covariance vector, this function returns the aggregated
    covariance matrix"""
    dim = len(var_vector)
    out = np.zeros((dim, dim))
    for i in range(dim):
        out[i, i] = var_vector[i]
        for j in range(i):
            cov = cov_vector[i - 1] * (-1 / damping) ** (i - j - 1)
            out[i, j] = cov
            out[j, i] = cov
    return out


def ijm_covariance(sigma, jump):
    """This function calculates the """
    dim = sigma.shape[0]
    out = np.zeros((dim, dim))
    for i in range(dim):
        for j in range(dim):
            out[i, j] = jump**2 * cov_ijm(sigma, i, j)
    return out


def cov_ijm(sigma, i, j):
    if i > j:
        return cov_ijm(sigma, j, i)
    var_comp = sum([sigma[k, k] for k in range(i+1)])
    cov_comp = 0
    for jj in range(i+1):
        for ii in range(i+1, j+1):
            cov_comp += sigma[ii, jj]

    return var_comp + cov_comp
```

### E.3.2 covariance_constructor.py

```
from typing import Callable


class CovarianceConstructor:
    """
    One of the parameters of the DA problem is Sigma. This class controls how this Sigma is created.

    ---> The covariance_method is the covariance structure of the meetings and is either

                         dmj_covariance or amj_covariance

```

```
        amj_covariance requires an additional argument, the default is phi=1,
        which gives independent jumps.

    ---> The aggregation method is the method for aggregating the correlations between meetings and is one of

                        tridiag_covariance or dta_covariance (independent jumps)

    """
    def __init__(self, covariance_method: Callable, aggregation_method: Callable):
        self.covariance_method = covariance_method
        self.aggregation_method = aggregation_method

```

# E.4 optimization_framework

## E.4.1 da_solution.py

```python
import numpy as np
from library.linalg import K, diag


class DurationSolution:
    """The normalized, average duration difference solution. All resulting vectors need to be scaled accordingly"""

    def __init__(self, base_solution: np.ndarray, differential: np.ndarray = None):
        """The base_solution is the optimal V in the DA problem. The differential is only used when solving the
        regular Markowitz problem"""
        self.x = base_solution
        self.eta = differential

        self.dim = base_solution.shape[0]

    def portfolio(self, lam: float = 0):
        """Portfolio is created based on the base solution and the differential, this provides the full Markowitz
        Bullet

                                !!!ONLY USED IN REGULAR MARKOWITZ PROBLEM!!!
        """

        if self.eta is None and lam != 0:
            raise ValueError("Solution Differential is not defined")

        if lam == 0:
            return self.x

        return self.x + lam * self.eta

    def lack(self, lam: float = 0, partition_norm: list = None):
        """From a solution V = Delta F, we calculate F = -K V"""
        if partition_norm is None:
            return K(self.dim) @ self.portfolio(lam)
```

```python
        return diag(partition_norm) @ K(self.dim) @ self.portfolio(lam)

    def duration(self, lam: float = 0, partition_norm: list = None):
        """From the lack F, we calculate the duration"""
        if partition_norm is None:
            return np.ones((self.dim, 1)) - self.lack(lam)
        return diag(partition_norm) @ self.duration(lam)

    def concentration(self):
        """Function for evaluating the concentration of a portfolio, not currently used."""
        return np.sum(self.x**2)

    def __str__(self):
        return f'Solution: {self.x} | Differential: {self.eta}'

```

### E.4.2 risk_reward.py

```python
from typing import Callable

import numpy as np

from portfolio_tool.duration_framework.duration_distribution import DurationDistribution
from portfolio_tool.market_framework.interest_model import TimePartition
from portfolio_tool.optimisation_framework.da_solution import DurationSolution


class MarkowitzBullet:
    """Class that holds the Markowitz bullet along with a list of the optimal portfolios"""
    duration_distribution: list[DurationDistribution] | list[DurationSolution]

    def __init__(self, risk: list | np.ndarray,
                 reward: list | np.ndarray,
                 duration_distribution: list[DurationSolution] = None,
                 exponential = False):
        self.risk = np.array(risk)
        self.reward = np.array(reward)
        if exponential:
            self.reward = np.exp(self.reward) - 1
        self.duration_distribution = duration_distribution

        self.len = self.risk.shape[0]

    def to_duration_distribution(self, time_partition: TimePartition):
        """Convert the optimal allocation in terms of V into the corresponding duration distribution
    """
        res = []
        sol: DurationSolution
        for sol in self.duration_distribution:
            distribution = DurationDistribution(sol.duration(partition_norm=time_partition.norm()).T
    [0], time_partition)
            res.append(distribution)
        self.duration_distribution = res

    def __getitem__(self, item):
        if item == 0:
```

```
            return self.risk
        if item == 1:
            return self.reward
        raise IndexError

    def get_vertical(self, item) -> (float, float, DurationDistribution):
        """Returns the (risk, reward, duration_distribution) triplet of a given index"""
        return self.risk[item], self.reward[item], self.duration_distribution[item]

    def get_max_reward(self) -> (float, float, DurationDistribution):
        """Returns the (risk, reward, duration_distribution) triplet corresponding to the portfolio with the
        biggest reward"""
        max_index = self.reward.argmax()
        return self.get_vertical(max_index)

    def get_min_risk(self) -> (float, float, DurationDistribution):
        """Returns the (risk, reward, duration_distribution) triplet corresponding to the portfolio with the
        lowest risk"""
        max_index = self.risk.argmin()
        return self.get_vertical(max_index)

    def get_index(self,
                  f1: Callable[[int, list], bool] = None,
                  f2: Callable[[int, list], bool] = None,
                  f3: Callable[[int, list], bool] = None,
                  first=True):
        """Returns the index of the first element that matches a certain criterion"""
        if (f1 is None) + (f2 is None) + (f3 is None) != 2:
            raise ValueError("Exactly one of f1, f2 and f3 must be Callable")

        return_lst = []
        if f1 is not None:
            f = f1
            lst = self.risk

        elif f2 is not None:
            f = f2
            lst = self.reward

        elif f3 is not None:
            f = f3
            lst = self.duration_distribution

        for i, el in enumerate(lst):
            if f(el, lst):
                if first:
                    return i
                return_lst.append(i)

        return return_lst
```

## E.5 duration_allocation

### E.5.1 markowitz.py

```python
import numpy as np
from library.linalg import mat_norm, quadratic_form, diag
from qpsolvers import solve_qp


class MarkowitzModel:
    """Class for the markowitz model"""
    def __init__(self, sigma: np.ndarray, M: np.ndarray):
        self.M = M
        self.sigma = sigma

        self.dim = self.sigma.shape[0]

    def solve(self):
        """Solve the standard markowitz problem"""
        one = np.ones((self.dim, 1))
        sigma_i = np.linalg.linalg.inv(self.sigma)
        a = mat_norm(one, sigma_i)
        b = quadratic_form(self.M, one, sigma_i)

        x_a = np.matmul(sigma_i, one) / a

        num = np.matmul(sigma_i, self.M - b * one / a)
        denom = mat_norm(self.M - b * one / a, sigma_i)

        eta = num / denom

        return x_a, eta / np.linalg.norm(eta)

    def constraint_solve(self, m, concentration_limit: list = None):
        """Solve the markowitz problem with the long-only constraint"""
        P = 2 * self.sigma
        q = np.zeros((self.dim, 1))

        if concentration_limit is None:
            G = np.zeros((1, self.dim))
            h = np.array([[0]])

        else:
            G = diag([1 for _ in range(self.dim)])
            h = np.array([concentration_limit]).T

        A = np.vstack([self.M.T, np.ones((1, self.dim))])

        b = np.array([[m], [1]])

        return solve_qp(P, q, G, h, A, b, solver="quadprog", lb=np.zeros((self.dim, 1)))

```

### E.5.2 da_solver.py

```python
import numpy as np

from library.linalg import mat_norm, diag, K
from portfolio_tool.duration_allocation.markowitz import MarkowitzModel
from portfolio_tool.duration_framework.duration_distribution import DurationDistribution
from portfolio_tool.market_framework.interest_model import InterestModel, TimePartition
from portfolio_tool.optimisation_framework.da_solution import DurationSolution
from portfolio_tool.optimisation_framework.risk_reward import MarkowitzBullet
from portfolio_tool.parameter_construction.covariance_constructor import CovarianceConstructor
from portfolio_tool.parameter_construction.covariance_functions import amj_covariance,
    dta_covariance


class DurationProblemParameters:
    """Class that holds the parameters for the duration allocation problem"""
    def __init__(self, sigma: np.array, ret: np.array, partition_norm: np.array):
        self.sigma = sigma
        self.ret = ret
        self.partition_norm = partition_norm

        self.dim = self.sigma.shape[0]

    def get_markowitz_params(self):
        """Transform the parameters for the duration allocation problem into parameters for the
    Markowitz Problem"""
        tr_sigma = mat_norm(self.partition_norm @ K(self.dim), self.sigma)
        tr_ret = (self.partition_norm @ K(self.dim)).T @ self.ret

        return tr_sigma, tr_ret


class DurationAllocation:
    """This class holds the duration allocation problem and facilitates is solver"""
    param_cache: DurationProblemParameters | None
    markowitz_cache: MarkowitzModel | None
    solution_cache: DurationSolution | None

    def __init__(self, interest_model: InterestModel):
        self.interest_model = interest_model
        self.dim = len(self.interest_model.rho)

        self.param_cache = None
        self.markowitz_cache = None
        self.solution_cache = None

    def initialize_parameters(self,
                              covariance_constructor: CovarianceConstructor=None,
                              beta: np.ndarray=None,
                              lam=0,
                              gamma=0,
                              amj_param=1,
                              dta_param=2):
        """Initialise the parameters of the DA problem. Hold the parameters in the parameter cache
    """
        if covariance_constructor is None:
            covariance_constructor = CovarianceConstructor(lambda x: amj_covariance(x, amj_param),
    lambda x, y: dta_covariance(x, y, dta_param))
        covariance_method, aggregation_method = covariance_constructor.covariance_method,
```

```
covariance_constructor.aggregation_method
        sigma = self.interest_model.covariance(covariance_method, aggregation_method)
        if beta is None:
            beta = self.interest_model.beta

        beta = np.array(beta).reshape((self.dim, 1))
        rho = np.array(self.interest_model.rho).reshape((self.dim, 1))
        ret = rho - beta

        normalizer = np.min(sigma.diagonal())

        sigma -= lam * normalizer * diag([1 for _ in range(self.dim)])
        sigma -= gamma * normalizer * diag(ret)

        self.param_cache = DurationProblemParameters(sigma, ret, diag(self.interest_model.
time_partition.norm()))

    def initialize_markowitz(self):
        """Initialize the markowitz problem using the parameters in the parameter cache"""
        if self.param_cache is None:
            raise ValueError("Parameter cache is empty")
        self.markowitz_cache = MarkowitzModel(*self.param_cache.get_markowitz_params())

    @staticmethod
    def initialize(time_partition: list,
                   initial_interest: float,
                   jump_size: float,
                   beta: list,
                   rho: list,
                   covariance_constructor: CovarianceConstructor = None,
                   amj_param=1,
                   dta_param=2):
        """Initialise an instance of the solver"""

        time_partition = TimePartition(time_partition)

        interest_model = InterestModel(time_partition, initial_interest, jump_size, beta, rho)
        foreign_exchange = DurationAllocation(interest_model)

        foreign_exchange.initialize_parameters(covariance_constructor, lam=0, amj_param=amj_param,
dta_param=dta_param)
        foreign_exchange.initialize_markowitz()

        return foreign_exchange

    def solve(self):
        """Solve the standard markowitz problem (without long-only constraint)"""
        if self.markowitz_cache is None:
            raise AttributeError("Markowitz Cache not initialized")
        solution, differential = self.markowitz_cache.solve()
        self.solution_cache = DurationSolution(solution, differential)
        return self.solution_cache

    def constraint_solve(self, m, concentration_limit: list = None):
        """Solve the Duration Allocation problem with reward parameter m and concentration limit if
desired"""
        if self.markowitz_cache is None:
            raise AttributeError("Markowitz Cache not initialized")
```

```
        I_norm = np.array([self.interest_model.time_partition.norm()]).T
        M = self.param_cache.ret
        return_param = (I_norm.T @ M)[0][0] - m
        self.solution_cache = DurationSolution(
            self.markowitz_cache.constraint_solve(return_param, concentration_limit=concentration_limit).reshape((self.dim, 1)))
        return self.solution_cache

    def risk_reward(self, lam: float = 0,
                    solution: DurationSolution = None,
                    scalar: float = 1,
                    vec: np.ndarray = None,
                    durations: DurationDistribution = None):
        """For a given duration allocation, return the risk and the reward. If no duration allocation is provided,
        the risk and reward of the allocation in the solution_cache are returned"""
        if solution is None:
            solution = self.solution_cache

        if vec is not None:
            solution = DurationSolution(vec)

        if durations is None:

            F = solution.lack(lam) * scalar
            D = solution.duration(lam) * scalar

        else:
            D = np.array([durations.durations]).T * scalar
            F = scalar - D

        I_m = np.array([self.interest_model.time_partition.norm()]).T

        risk = mat_norm(diag(I_m)@F, self.param_cache.sigma)[0][0]
        reward = ((diag(I_m) @ D).T @ self.param_cache.ret)[0][0]
        return risk, reward

    def parametric_solution(self,
                            lam_range: [int, int],
                            amplifier: float = 1,
                            scalar: float = 1,
                            std_dev=False):
        risk_lst = []
        ret_lst = []
        for lam in range(*lam_range):
            lam *= amplifier
            risk, reward = self.risk_reward(lam, scalar=scalar)
            if std_dev:
                risk = np.sqrt(risk)
            risk_lst.append(risk)
            ret_lst.append(reward)

        return MarkowitzBullet(risk_lst, ret_lst)

    def feasible_set(self, concentration_limit: list = None):
        """Determine the feasible set of the DA problem with the given concentration limit"""
        I = self.interest_model.time_partition.norm()
        M = self.param_cache.ret
```

```python

        if concentration_limit is None:
            concentration_limit = [1 for _ in range(self.dim)]

        H = [I[i] * M[i, 0] * concentration_limit[i] for i in range(len(I))]
        X = [sum(H[:i]) for i in range(len(H))]

        return min(X), max(X)

    def full_constraint_solve(self, N: int = 100,
                              concentration_limit: list = None,
                              print_error=False):
        """Solve the DA problem for the full (discretised) feasible set"""
        sol_lst = []
        min_m, max_m = self.feasible_set(concentration_limit=concentration_limit)

        for i in range(N + 1):
            m = i / N * (max_m - min_m) + min_m
            try:
                sol = self.constraint_solve(m, concentration_limit=concentration_limit)
            except Exception as error:
                if print_error:
                    print(f'Failure: {i}: {error}')
                continue
            sol_lst.append(sol)

        return sol_lst

    def risk_reward_curve(self,
                          N: int = 100,
                          concentration_limit: list = None,
                          scalar: float = 1,
                          exponential=False,
                          std_dev=False):
        """Returns the Markowitz Bullet of the DA problem with the given concentration limit"""
        sol_lst = self.full_constraint_solve(N, concentration_limit)
        risk_lst = []
        reward_lst = []

        for sol in sol_lst:
            risk, reward = self.risk_reward(solution=sol, scalar=scalar)
            if std_dev:
                risk = np.sqrt(risk)
            risk_lst.append(risk)
            reward_lst.append(reward)

        risk_reward = MarkowitzBullet(risk_lst, reward_lst, sol_lst, exponential=exponential)
        risk_reward.to_duration_distribution(self.interest_model.time_partition)

        return risk_reward

    # def get_portfolio_from_risk(self, risk: float, lam_range: int, amp: float = 1):
    #     prev_risk = -1
    #
    #     for lam in range(lam_range):
    #         lam *= amp
    #         current_risk, _ = self.risk_reward(lam)
    #
```

```
    #         if prev_risk <= risk <= current_risk:
    #             return lam
    #
    #         prev_risk = current_risk
    #
    #     return None

    # def get_portfolio_from_reward(self, reward: float, lam_range: int, amp: float = 1):
    #     prev_reward = -1
    #
    #     for lam in range(lam_range):
    #         lam *= amp
    #         _, current_reward = self.risk_reward(lam)
    #
    #         if prev_reward <= reward <= current_reward:
    #             return lam
    #
    #         prev_reward = current_reward
    #
    #     return None

```

## E.6 control

### E.6.1 controller.py

```
import matplotlib.pyplot as plt
from pylab import mpl
import numpy as np
import pandas as pd

from library.constants import IMAGE_PATH, PATH
from library.funcs import force_zeros
from portfolio_tool.duration_allocation.da_solver import DurationAllocation
from portfolio_tool.duration_framework.bond_investment import BondInvestment
from portfolio_tool.duration_framework.duration_distribution import DurationDistribution


class Controller:
    """Central controller to solve duration allocation problems"""
    def __init__(self, exchange: DurationAllocation):
        self.exchange = exchange

    def benchmark(self, scalar):
        benchmark_portfolio = BondInvestment.from_dict({90: 1}, self.exchange.interest_model)
        benchmark_duration = DurationDistribution(benchmark_portfolio.get_durations(average=True,
    relative=True),
                                                  self.exchange.interest_model.time_partition)

        risk, reward = self.exchange.risk_reward(durations=benchmark_duration)
        cost = benchmark_portfolio.get_payoff()*scalar

        return cost, risk*scalar**2, reward*scalar

    def get_triple_curve(self,
```

```
                         concentration_limit: list,
                         N: int = 100,
                         lam_range: list = None,
                         scalar: float = 1,
                         exponential=False,
                         std_dev=False):
        """Returns the markowitz bullet for the standard markowitz problem, the DA problem and
        the concentration limited problem."""
        if lam_range is None:
            lam_range = [-10, 11]

        self.exchange.solve()
        markowitz = self.exchange.parametric_solution(lam_range, scalar=scalar, std_dev=std_dev)
        long_only = self.exchange.risk_reward_curve(N, scalar=scalar, exponential=exponential, std_dev=std_dev)
        constrained = self.exchange.risk_reward_curve(N, concentration_limit, scalar, exponential=exponential, std_dev=std_dev)

        return markowitz, long_only, constrained

    def plot_triple_curve(self,
                          concentration_limit: list,
                          N: int = 100,
                          lam_range: list = None,
                          scalar: float = 1,
                          y_lim: tuple = None,
                          plot=True,
                          exponential=False,
                          benchmark=False):
        """Plots the markowitz bullet for the standard markowitz problem, the DA problem and
        the concentration limited problem."""

        markowitz, long_only, constrained = self.get_triple_curve(concentration_limit,
                                                                   N,
                                                                   lam_range,
                                                                   scalar,
                                                                   exponential=exponential)

        if y_lim is None:
            y_lim = (min(long_only.reward), max(long_only.reward))

            if plot:
                plt.figure(figsize=(10, 3))

            plt.plot(*markowitz, label='Markowitz Bullet')
            plt.plot(*long_only, label='Long-Only')
            plt.plot(*constrained, label='Concentration Limited')

            if benchmark:
                _, bm_risk, bm_reward = self.benchmark(scalar)
                plt.plot(bm_risk, bm_reward, 'ko', label='3M Benchmark')

            plt.ylim(y_lim)
            plt.legend()

            if plot:
                plt.show()

```

```python
    def plot_double_curve(self,
                          concentration_limit: list,
                          N: int = 100,
                          lam_range: list = None,
                          scalar: float = 1,
                          y_lim: tuple = None,
                          plot=True,
                          exponential=False,
                          std_dev=False,
                          benchmark=False,
                          filename=None):
        """Plots the markowitz bullet for the DA problem and the concentration limited problem."""

        markowitz, long_only, constrained = self.get_triple_curve(concentration_limit,
                                                                  N,
                                                                  lam_range,
                                                                  scalar,
                                                                  exponential=exponential,
                                                                  std_dev=std_dev)


        if plot:
            plt.figure(figsize=(6, 3))

        plt.plot(*long_only, label='Duration Allocation')
        plt.plot(*constrained, label='Concentration Limited')

        if benchmark:
            _, bm_risk, bm_reward = self.benchmark(scalar)
            if std_dev:
                bm_risk = np.sqrt(bm_risk)
            plt.plot(bm_risk, bm_reward, 'ko', label='3M Benchmark')

        plt.ylim(y_lim)
        plt.legend()

        plt.xlabel('Risk')
        plt.ylabel('Reward Compared to Market Expectation')

        plt.tight_layout()

        if plot:
            if filename:
                plt.savefig(PATH / 'output' / 'plots' / f'{filename}_Reward.pdf')
            plt.show()


    def make_bullet_array(self,
                         concentration_limit: list,
                         N: int = 100,
                         scalar: float = 1,
                         exponential=False):

        bullet = self.exchange.risk_reward_curve(N, concentration_limit, scalar, exponential=
    exponential)
        out = []
        for i in range(bullet.len):
            dur: DurationDistribution
```

```
            risk, rew, dur = bullet.get_vertical(i)
            row = [risk, rew] + list(dur.durations)
            out.append(row)
        return out

    def make_bullet_df(self,
                         concentration_limit: list,
                         N: int = 100,
                         scalar: float = 1,
                         exponential=False):

        df = pd.DataFrame(self.make_bullet_array(concentration_limit, N, scalar, exponential),
                          columns=['Risk', 'Reward'] + [f'Period {i+1}' for i in range(len(concentration_limit))])
        return df

    def make_optimal_portfolio_array(self,
                         concentration_limit: list,
                         N: int = 100,
                         scalar: float = 1,
                         exponential=False):

        bullet = self.exchange.risk_reward_curve(N, concentration_limit, scalar, exponential=exponential)
        out = []
        for i in range(bullet.len):
            dur: DurationDistribution
            risk, rew, dur = bullet.get_vertical(i)
            row = [force_zeros(np.sqrt(risk), 2), force_zeros(rew, 2)] + list(dur.diff()) + [0]
            out.append(row)
        return out

    def make_optimal_portfolio_df(self,
                                  concentration_limit: list,
                                  N: int = 100,
                                  scalar: float = 1,
                                  exponential=False):

        df = pd.DataFrame(self.make_optimal_portfolio_array(concentration_limit, N, scalar, exponential),
                          columns=['Risk', 'Reward'] + [f'Period {i+1}' for i in range(len(concentration_limit))])
        return df

    def min_risk_portfolio(self, concentration_limit: list) -> (float, float, BondInvestment):

        risk_reward = self.exchange.risk_reward_curve(concentration_limit=concentration_limit)
        risk, rew, durations = risk_reward.get_min_risk()
        swap_portfolio = durations.vertical_bond_investment(interest_model=self.exchange.interest_model)

        return risk, rew, swap_portfolio

    def get_risk_cost_curves(self,
                             concentration_limit: list,
                             N: int = 100,
                             scalar: float = 1,
                             exponential=False,
```

```
                             std_dev=False):
        _, da_curve, conc_curve = self.get_triple_curve(concentration_limit,
                                                        N,
                                                        None,
                                                        scalar,
                                                        exponential,
                                                        std_dev)

        da_costs = []
        conc_costs = []

        for dur_distr in da_curve.duration_distribution:
            da_costs.append(scalar*dur_distr.flat_payoff(self.exchange.interest_model.beta,
                                                         self.exchange.interest_model.rho))

        for dur_distr in conc_curve.duration_distribution:
            conc_costs.append(scalar*dur_distr.flat_payoff(self.exchange.interest_model.beta,
                                                           self.exchange.interest_model.rho))

        return [da_curve.risk, da_costs], [conc_curve.risk, conc_costs]

    def plot_risk_cost_curve(self,
                             concentration_limit: list,
                             N: int = 100,
                             scalar: float = 1,
                             y_lim: tuple = None,
                             plot=True,
                             exponential=False,
                             std_dev=False,
                             benchmark=False,
                             filename=None):
        """Plots the markowitz bullet for the DA problem and the concentration limited problem."""

        da, conc = self.get_risk_cost_curves(concentration_limit,
                                             N,
                                             scalar,
                                             exponential=exponential,
                                             std_dev=std_dev)

        if plot:
            plt.figure(figsize=(6, 3))
        plt.plot(*da, label='Duration Allocation')
        plt.plot(*conc, label='Concentration Limited')

        if benchmark:
            bm_cost, bm_risk, _ = self.benchmark(scalar)
            if std_dev:
                bm_risk = np.sqrt(bm_risk)
            plt.plot(bm_risk, bm_cost, 'ko', label='3M Benchmark')

        plt.ylim(y_lim)
        plt.legend()
        # plt.gca().yaxis.set_major_formatter(mpl.ticker.StrMethodFormatter('{x:.0f}'))
        plt.xlabel("Risk")
        plt.ylabel("Return")

        plt.tight_layout()
        if plot:
```

```python
            if filename:
                plt.savefig(PATH / 'output' / 'plots' / f'{filename}_Return.pdf')
            plt.show()


    # region: Deprecated (Potential future features)
    def payoff_return_curves(self,
                             concentration_limit: list,
                             N: int = 100,
                             lam_vals: list | float = None,
                             scalar: float = 1):
        """Plots for evaluating expected returns based on estimator accuracy
        Not used"""

        if lam_vals is None:
            lam_vals = [-0.5, 0.5, 1, 1.5]

        risk_reward_curves = self.exchange.risk_reward_curve(N, concentration_limit, scalar=scalar)
        risk_lst = risk_reward_curves.risk
        reward_lst = risk_reward_curves.reward
        sol_lst = risk_reward_curves.duration_distribution

        if type(lam_vals) in [int, float]:
            curve_list = []

            for sol in sol_lst:
                sol: DurationDistribution
                curve_list.append(sol.lambda_payoff_list(self.exchange.interest_model,
                                                         capital=scalar,
                                                         lam=lam_vals))

            curve_list = np.array(curve_list).T
            return risk_lst, reward_lst, curve_list

        plot_list = []
        for lam in lam_vals:
            _, _, curve_list = self.payoff_return_curves(concentration_limit, N, lam, scalar)
            plot_list.append(curve_list)
        return risk_lst, reward_lst, plot_list

    def make_payoff_return_plotters(self,
                                    concentration_limit: list,
                                    N: int = 100,
                                    lam_vals: list | float = None,
                                    scalar: float = 1):
        """Plots for evaluating expected returns based on estimator accuracy
        Not used"""

        legend = ['Flat', 'Benchmark', 'Full', 'Even']
        risk_lst, reward_lst, curves = self.payoff_return_curves(concentration_limit, N, lam_vals,
    scalar)

        if lam_vals is None:
            lam_vals = [-0.5, 0.5, 1, 1.5]

        plot_size = 200 + (len(lam_vals)//2 + (len(lam_vals)%2 != 0)) * 10 + 1

        def plot(x_lst, filename=None):
```

```python
            if type(lam_vals) is list:
                plt.figure(figsize=(10, 6))
                plt.tight_layout()
                for i, arr in enumerate(curves):
                    max_val = np.max(arr[:-1])
                    min_val = np.min(arr[:-1])

                    diff = max_val - min_val
                    plt.subplot(plot_size + i)
                    plt.title(f"Lambda = {lam_vals[i]}")
                    for j, lst in enumerate(arr):
                        plt.plot(x_lst, lst, label=legend[j])

                    plt.ylim(min_val - diff/2, max_val+diff/2)

                    plt.legend()
                if filename is not None:
                    plt.savefig(IMAGE_PATH / f'{filename}.png')
                plt.show()
                return

            plt.figure(figsize=(16, 8))
            max_val = np.max(curves[:-1])
            min_val = np.min(curves[:-1])

            diff = max_val - min_val
            plt.title(f"Lambda = {lam_vals}")
            for j, lst in enumerate(curves):
                plt.plot(x_lst, lst, label=legend[j])
            plt.legend()
            plt.ylim(min_val - diff/2, max_val + diff/2)
            if filename is not None:
                plt.savefig(IMAGE_PATH / f'{filename}.png')
            plt.show()

        return lambda filename=None: plot(risk_lst, filename), lambda filename=None: plot(reward_lst
    , filename)

    def concentration_return_curve(self,
                                   concentration_limit: list,
                                   N: int = 100,
                                   scalar: float = 1):
        """Plots for evaluating expected returns based on estimator accuracy
        Not used"""

        risk_reward = self.exchange.risk_reward_curve(N, concentration_limit, scalar)

        duration_list = risk_reward.duration_distribution

        concentration_list = []

        for duration in duration_list:
            duration: DurationDistribution
            semi_even = duration.semi_even_bond_investment(self.exchange.interest_model, capital=
    scalar)
            concentration_list.append(semi_even.concentration())

        return risk_reward.reward, concentration_list
```

```

    def max_reward_correcteness_curve(self,
                                      concentration_limit: list,
                                      N: int = 100,
                                      N_lam: int = 100,
                                      lam_range: list = None,
                                      scalar: float = 1,
                                      plot = False):

        sol = self.exchange.risk_reward_curve(N, concentration_limit, scalar)
        duration: DurationDistribution
        _, _, duration = sol.get_max_reward()

        lam_lst, curves = duration.correctness_curves(self.exchange.interest_model, capital=scalar, lam_range=lam_range, N=N_lam)

        if plot:
            duration.plot_correctness_curves(self.exchange.interest_model, scalar, lam_range, N, plot=True)

        return lam_lst, curves

    def min_risk_correcteness_curve(self,
                                      concentration_limit: list,
                                      N: int = 100,
                                      N_lam: int = 100,
                                      lam_range: list = None,
                                      scalar: float = 1,
                                      plot=False):
        """Plots for evaluating expected returns based on estimator accuracy
        Not used"""

        sol = self.exchange.risk_reward_curve(N, concentration_limit, scalar)
        duration: DurationDistribution
        _, _, duration = sol.get_min_risk()

        lam_lst, curves = duration.correctness_curves(self.exchange.interest_model, capital=scalar, lam_range=lam_range,
                                                      N=N_lam)

        if plot:
            duration.plot_correctness_curves(self.exchange.interest_model, scalar, lam_range, N, plot=True)

        return lam_lst, curves

    # endregion
```

## E.7 simulation

### E.7.1 simulator.py

```
import numpy as np

from portfolio_tool.control.controller import Controller
```

```python
from portfolio_tool.duration_allocation.da_solver import DurationAllocation
from portfolio_tool.parameter_construction.covariance_constructor import CovarianceConstructor
from portfolio_tool.parameter_construction.covariance_functions import amj_covariance,
    dta_covariance

from tqdm import tqdm

class Simulator:
    def __init__(self,
                 time_partition: list,
                 initial_interest: float,
                 jump_size: float,
                 capital: int = 1):
        self.time_partition = time_partition
        self.initial_interest = initial_interest
        self.jump_size = jump_size
        self.capital = capital

        self.cov_constr = CovarianceConstructor(lambda x: amj_covariance(x, 2),
                                                dta_covariance)

        self.rho = None
        self.beta = None

        self.benchmark1 = np.array([[0 for _ in range(len(time_partition))]]).T
        self.benchmark1[4] = 1

        self.benchmark2 = np.array([[0 for i in range(len(time_partition))]]).T
        self.benchmark2[7] = 1

    def construct_markets(self, rho_opt: list, beta_opt: list):
        rho_lst = [[]]
        beta_lst = [[]]

        for i in range(len(self.time_partition) - 1):
            rho_temp = rho_lst.copy()
            rho_lst = []

            beta_temp = beta_lst.copy()
            beta_lst = []

            for j in (tqdm(range(len(rho_temp)))):
                market_rho = rho_temp[j]
                market_beta = beta_temp[j]
                for k in range(len(rho_opt)):
                    current_rho = 0 if i == 0 else market_rho[-1]
                    current_beta = 0 if i == 0 else market_rho[-1]

                    new_rho = market_rho + [current_rho + (-1) ** i * rho_opt[k]]
                    new_beta = market_beta + [current_beta + (-1) ** i * beta_opt[k]]

                    rho_lst.append(new_rho)
                    beta_lst.append(new_beta)

        for rho, beta in zip(rho_lst, beta_lst):
            rho.append(rho[-1] + 0.05)
            beta.append(rho[-1] + 0.05)

```

```python
        self.rho = rho_lst
        self.beta = beta_lst

    def run(self):
        risk_lst = []
        reward_lst = []

        dur_lst = []

        b1_risk_lst = []
        b2_risk_lst = []

        b1_rew_lst = []
        b2_rew_lst = []

        faulty_sequences = 0

        for i in (tqdm(range(len(self.rho)))):

            foreign_exchange = DurationAllocation.initialize(self.time_partition,
                                                             self.initial_interest,
                                                             self.jump_size,
                                                             self.beta[i],
                                                             self.rho[i],
                                                             self.cov_constr)

            controller = Controller(foreign_exchange)

            try:
                risk_reward = controller.exchange.risk_reward_curve()
                risk, reward, dur = risk_reward.get_min_risk()
            except:
                faulty_sequences += 1
                continue

            b1_risk, b1_rew = controller.exchange.risk_reward(vec=self.benchmark1)
            b2_risk, b2_rew = controller.exchange.risk_reward(vec=self.benchmark2)

            risk_lst.append(risk)
            reward_lst.append(reward)

            dur_lst.append(dur)

            b1_risk_lst.append(b1_risk)
            b2_risk_lst.append(b2_risk)

            b1_rew_lst.append(b1_rew)
            b2_rew_lst.append(b2_rew)

        return [risk_lst, reward_lst, dur_lst, b1_risk_lst, b2_risk_lst, b1_rew_lst, b2_rew_lst]
```