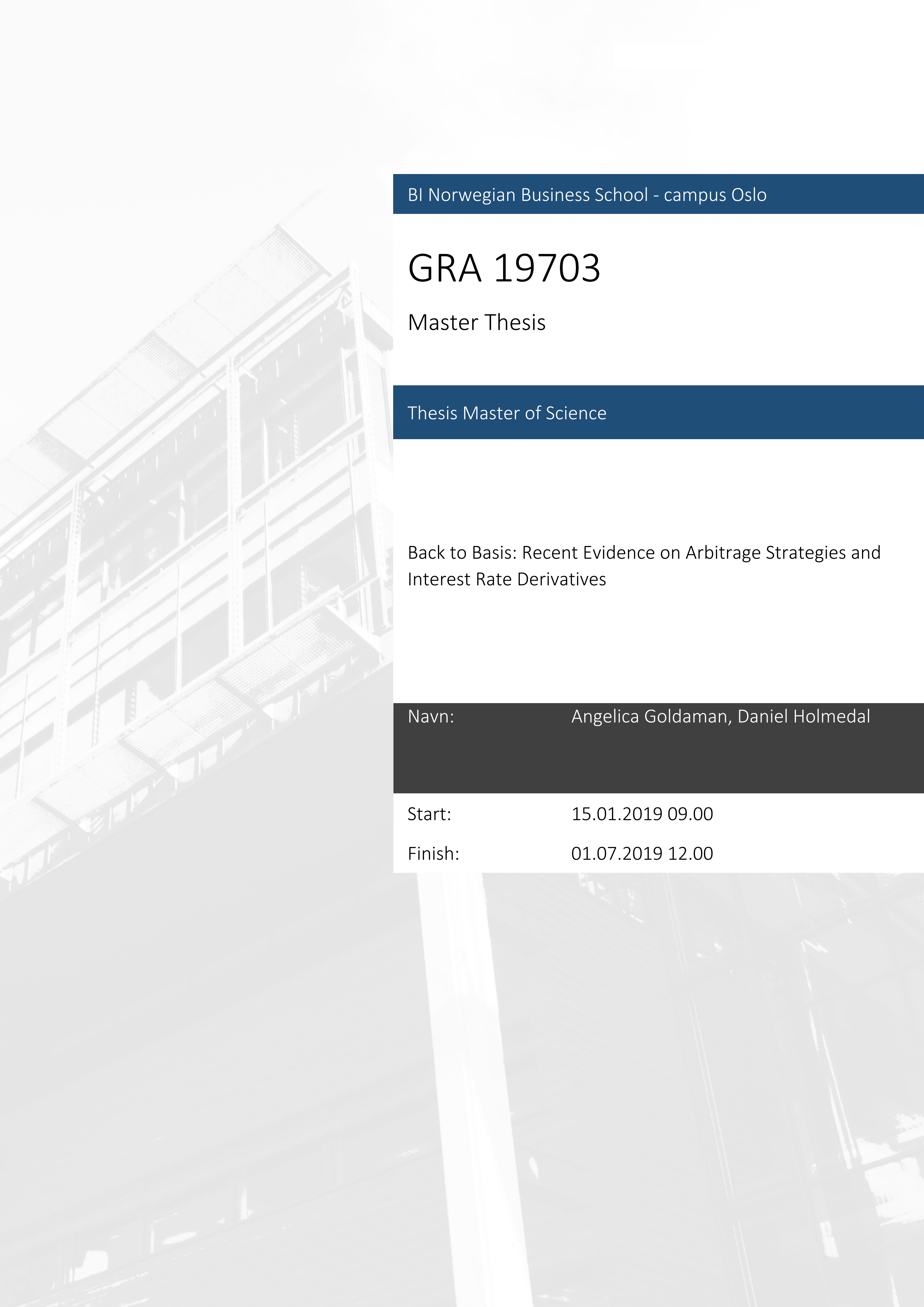


BI Norwegian Business School - campus Oslo

GRA 19703

Master Thesis

Thesis Master of Science

Back to Basis: Recent Evidence on Arbitrage Strategies and Interest Rate Derivatives

Navn: Angelica Goldaman, Daniel Holmedal

Start:	15.01.2019 09.00
Finish:	01.07.2019 12.00

MSc Thesis - BI Norwegian Business School

Back to Basis

Recent Evidence on Arbitrage Strategies and Interest Rate Derivatives

Authors:

Angelica Goldaman
MSc in Business, Major in Finance

Daniel Holmedal
MSc in Business, Major in Finance and QTEM

Supervisor:
Sven Klingler

July, 2019

Abstract

We examine near-arbitrage strategies in the market for interest rate derivatives. Using futures and forward rate agreements, we construct replication portfolios that match cash flows of vanilla interest rate swaps. Standard arbitrage theory suggests that the difference, or basis, between swap rates implied from futures and forward rate agreements and the market swap rate should be close to zero. Despite being some of the largest and most liquid markets in the world, we find mispricings using both futures and forward rate agreements.

This thesis is a part of the MSc programme at BI Norwegian Business School. The school takes no responsibility for the methods used, results found, or conclusions drawn.

Contents

List of Figures

List of Tables

1 Introduction

We examine near-arbitrage strategies in the market for interest rate derivatives. Using futures and forward rate agreements, we construct replication portfolios that match cash flows of vanilla interest rate swaps. Standard arbitrage theory suggests that the difference, or basis, between swap rates implied from futures and forward rate agreements and the market swap rate should be close to zero.

Despite being some of the largest and most liquid markets in the world, we find mispricings using both futures and forward rate agreements.

Our contribution to existing literature is three-fold. First, we use futures and forward rate agreements to provide a method for replicating cash flows of an interest rate swap in a post-crisis framework. Secondly, we use market prices of forward rate agreements as opposed to theoretical prices to price interest rate swaps. And lastly, we provide an overview of bases using both futures and forward rate agreements for USD-, EUR- and GBP-linked interest rate swaps.

The thesis is structured as follows. We start with a tour of relevant literature, highlighting previous research and underlining important changes in the valuation of interest rate swaps after the 2007-2008 financial crisis. Later, we provide an overview of relevant theory which includes characteristics of interest rate derivatives such as futures, forward rate agreements, and interest rate swaps. We construct replication portfolios showing how markets of futures, forward rate agreements and interest rate swaps are intimately linked. Motivated by these replication portfolios, we provide a flexible pricing framework that works across currencies. After that, we present an overview of the bases in all three currencies for both futures and forward rate agreements. We highlight differences before and after the financial crisis, and look at correlations between countries. Lastly, we highlight factors limiting arbitrage for two apparent arbitrage opportunities before we conclude the thesis.

2 Literature Review

The use of futures contracts to price interest rate swaps garnered much attention in the 1990's and early 2000's. The empirical treatment of the futures basis, defined as the difference between a futures implied swap rate and a market implied swap rate, has mostly focused on credit risk, liquidity and convexity as factors to explain the basis. Minton (1997) investigates the differential between futures implied swap rates and market swap rates, and claims the basis is caused by credit risk in interest rate swaps. In contrast, Bomfim (2003) finds that credit risk should not have a significant role on the futures basis, even in times of market stress. Both futures contracts and interest rate swaps are collateralized (Johannes & Sundaresan, 2007), and several empirical examinations assert that posting of collateral close to eliminates credit risk (Collin-Dufresne & Solnik, 2001; Liu, Longstaff, & Mandell, 2006; Feldhütter & Lando, 2008). Gupta and Subrahmanyam (2000) extend Minton's analysis to include credit risk, liquidity, asymmetric information and convexity. They disagree with Minton and find that although credit risk is statistically significant it is unlikely to be economically significant. They introduce convexity as a factor explaining away most of the basis.

Research has accumulated over the years showing the relationship between futures and future prices. A key difference is the way in which futures contracts are marked-to-market. A futures contract involves both an initial margin and a variation margin that is updated daily based on interest rate movements. This difference between futures rates and forward rates, often called convexity, received much attention from researchers in the 1980's and 1990's. Jarrow and Oldfield (1981) and Cox, Ingersoll and Ross (1981) were two of the first empirical treatments of the differences between futures and forward rates. Grinblatt and Jegadeesh (1996) show how theoretical futures and forward rates can be computed using term-structure models based on Vasicek (1977) and Cox, Ingersoll and Ross (2005). The goal of most these papers is to construct theoretical models that reconcile the mark-to-market features of a futures contract with the non-mark-to-market features of forward rates. At a general level, models for convexity adjustments can be grouped in two: equilibrium models and no-arbitrage models (Hull, 2018). Vasicek (1977) and Cox, Ingersoll, and Ross (2005) are equilibrium models that both provide closed-form solutions to the convexity bias. Hull and White (1990) is an example of a widely used no-arbitrage model, and its calibration involves fitting the current term-structure to a trinomial tree (Hull & White, 1994).

Pricing of interest rate swaps has gone through several changes in the decade following the 2007-2008 financial crisis. Mercurio (2009) and Ametrano & Bianchetti (2013) provide overviews of the most important changes following the crisis. Important considerations includes specific tenor considerations when constructing forward curves. Another consideration is the discounting curve. Johannes and Sundaresan (2007) claimed that collateralized instruments such as interest rate swaps should be discounted by a rate lower than interbank offered rates (IBORs). After the financial crisis, it has become commonly accepted to discount cash flows without credit risk using OIS rates rather than IBORs (Hull & White, 2013). Prior to the financial crisis, valuation of interest rate swaps could be done through the construction of a single curve. In the post-crisis environment, valuations need to be conducted using multiple curves (Veronesi, 2016).

A central concept in finance is arbitrage. If the cash flows following from two financial instruments are equal, they should have the same price. If prices deviate, an arbitrageur would sell the most expensive instrument and buy the least expensive. Shleifer and Vishny's (1997) seminal paper introduced limits to arbitrage, showing how obvious arbitrage opportunities became unprofitable once practical considerations were included. Duarte, Longstaff and Yu (2006) provide an overview of fixed-income arbitrage strategies and show that while some strategies are profitable, their profits is a compensation of risk in the strategy. Boyarchenko, Eisenbach, Gupta, Shachar, and Van Tassel (2018) provide an overview of post-crisis regulations and show how small arbitrage opportunities quickly become unprofitable.

3 Theory

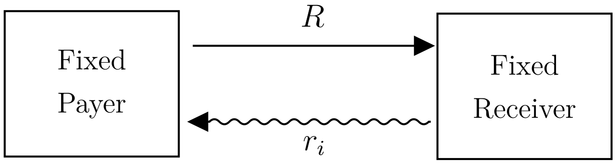
3.1 Interest Rate Derivatives

The market for interest rate derivatives is one of the largest financial markets in the world. Statistics from the Bank for International Settlements (BIS) estimates its size at the end of 2018 to be close to $450 trillion, with most contracts being denominated in USD or EUR (Bank for International Settlements, 2019). This market has grown exponentially since interest rate derivatives were introduced in the early 1980s. Interest rate derivatives are mainly used for hedging exposure to interest rate risk or for speculating on future interest rate movements.

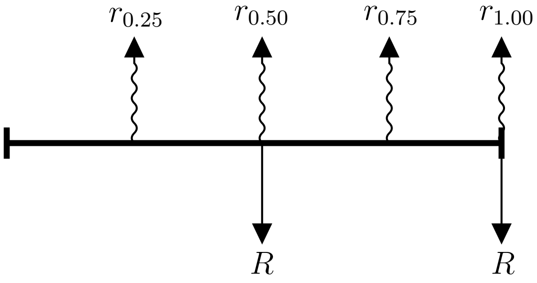
Most interest rate derivatives are indexed against an interbank offered rate (IBOR) which is the rate of interest on short-term loans between large banks. In 2013, LIBOR was used as the reference rate in more than $300 trillion of financial contracts (Hou & Skeie, 2014). Below, we walk through some of the most popular types of interest rate derivatives.

3.1.1 Interest Rate Swaps (IRSs)

An interest rate swap is an agreement between two parties in which one party agrees to pay a fixed rate, R, known as the swap rate, and the other pays a floating rate, r_i.



Payments happen at agreed upon dates in the future. For USD interest rate swaps, the most common payment frequency is semiannual for the fixed leg and quarterly for the floating leg. For other currencies, the payment frequencies are different. In the case of a one-year USD IRS, the payments would occur at the following times

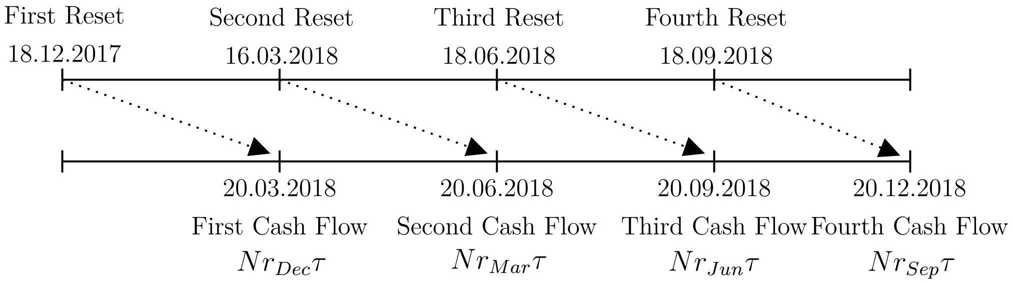


The floating rate is usually based on an IBOR rate. For USD swaps, it is common for the floating payments to be equal to the three-month LIBOR rate prevailing three months before the floating payment. In a sense, the floating rate becomes fixed for three months. The dates at which the floating rates are fixed are called reset dates.

Consider a one-year USD IRS running from December 20th, 2017 to December 20th, 2018. In the timelines below, we see that the floating payment on March 20th, 2018 will be fixed to the three-month LIBOR rate trading at December 18th, 2017[1]. Let us denote the three-month LIBOR in December as r_{Dec}. The floating

[1] It is common for both USD and EUR IBORs to settle two days after their trade date. This

payment in March will be the notional, N, of the swap multiplied by a LIBOR rate fixed three months ago, r_{Dec}, and finally multiplied by a year-fraction, τ, which is the actual number of days (also known as the accrual period) between December 20th, 2017 and March 20th, 2018 divided by 360.



The date schedule is key to understanding interest rate swaps. This schedule determines when payments from each leg are made, the accrual periods for the rates, and the reset dates. There are other details to consider as well, but we get back to those later.

The market for interest rate swaps has adapted to several large regulatory changes throughout the years. Developments like the ISDA Master Agreement, first introduced in the mid-1980s, have tried to reduce the default risk in interest rate swap transactions. The ISDA Master Agreement is a legal framework that parties involved in a swap transaction agree to follow. It is of particular relevance for over-the-counter swaps. A key element in the agreement deals with collateralization. When the value of the swap contract becomes negative for one of the counterparties, they need to post collateral. This reduces the risk of this counterparty not meeting future payments, thereby reducing the default risk.

3.1.2 Forward Rate Agreements (FRAs)

A forward rate agreement is an interest derivative where the difference between a prevailing market rate and the forward rate is cash-settled at some point in the future. Consider a trader buying the 3x6 forward rate agreement at the rate r_{3x6}^{FRA} and with notional N. The number of days in the accrual period (i.e. between 3- and 6-months) divided by 360 is given by τ[2]. Denoting the IBOR rate in three

rate would settle on December 20th, 2017 and cover the period until March 20th, 2018. For GBP IBORs, settlement is usually on the same date as the trade date.

[2] This day-count convention holds for USD and EUR where ACT/360 is used. For GBP, ACT/365 is used.

months time as r_{3x6}^{IBOR} , the value of the FRA in three months time, and the amount that will be cash-settled can be written as

$$V_{FRA,3} = N\frac{(r_{3x6}^{IBOR} - r_{3x6}^{FRA})\tau}{1 + r_{3x6}^{IBOR}\tau}$$

FRAs enable us to lock in implied three-month IBOR rates in the future. Conveniently, their date schedules tend to align well with the date schedules of interest rate swaps. This allows users of IRSs to hedge their floating payments by entering into consecutive positions in FRAs. For an IRS with quarterly payments with floating payments fixed three months in advance, the 0x3 FRA can be used to hedge the first payment, and the 3x6 FRA can be used to hedge the second payment.

FRAs are available for most large currencies. For USD, EUR and GBP, the data goes back to the early 2000's. The liquidity in these contracts vary up until 2007/2008, after which they have become more popular instruments for users of interest rate derivatives.

In contrast to IRSs, credit risk is a more pressing concern when using FRAs. Collateralization is not as prevalent so positions in FRAs entail default risk.

3.1.3 Short-Term Interest Rate Futures (STIR-Futures)

STIR-futures are similar to FRAs insofar allowing us to lock in three-month rates starting in the future. Being a futures contract, they are more standardized than FRAs. As a result, their credit risk is minimal and more similar to a collateralized IRS. A potential drawback of this standardization involves date schedules that rarely align perfectly with date schedules of IRSs. Hence, hedging floating payments in an IRS with STIR-futures is not straightforward. Nevertheless, STIR-futures remain popular instruments for hedgers and speculators.

STIR-futures contracts linked to USD-, EUR- and GBP-interest rates are called Eurodollar futures, EURIBOR futures and Short Sterling futures respectively. The markets for Eurodollar futures and EURIBOR futures are the two largest, and they regularly trade in excess of one trillion \$ and € each day (Aikin, 2012). The Eurodollar futures market deserves particular attention. It is one of the deepest markets in the world, both in terms of volume and contract length. CME Group allows trades in 40 Eurodollar contracts, allowing users to lock in forward rates between IMM dates[3] for ten years into the future. This volume has historically given STIR-futures markets a clear advantage over FRAs, and it has long been common the express forward curves based of STIR-futures.

STIR-futures are derived from three-month interest rates. Prices of the contracts are given as 100 minus the expectation of three-month IBOR at expiration of the contract.

$$p^{STIR} = 100 - r^{STIR}$$

STIR-futures expire on the third Wednesday in the months March, June, September and December. On these dates, contracts are cash-settled against the prevailing three-month IBOR.

Consider an example where we buy the March STIR-futures contract (i.e. the contract expiring in March) on February 25th, 2019 at price p^{STIR}_{Mar}, essentially locking in an implied forward rate from March to June denoted by r^{STIR}_{Mar}. This contract will expire two business days[4] before the third Wednesday in the next IMM-month, which happens to be in March. At this date, March 18th, 2019, the final value is determined as 100 minus the three-month IBOR on that day, r^{IBOR}_{Mar}. Two days later, on the value date which in this case is March 20th, 2019, the contract is cash-settled for a final value of

$$V_{Mar} = N\tau^{STIR}\left(r^{IBOR}_{Mar} - r^{STIR}_{Mar}\right)$$

where N is the notional of the contract[5] and τ^{STIR} is the year-fraction for the accrual period of the forward rate. The accrual period for Eurodollar, EURIBOR and Short Sterling futures is 1/4 (OpenGamma, 2013).

Similarly as in an FRA, STIR-futures allow us to fix future three-month rates in advance. For a swap in which the rates of the floating leg are fixed three-months in advance, a March STIR-futures contract can be used to hedge the quarterly payment in June. To lock in an implied forward rate from June to September, we buy the June STIR-futures contract. This procedure of buying several STIR-futures contract with expiries after each other is often referred to as creating strips.

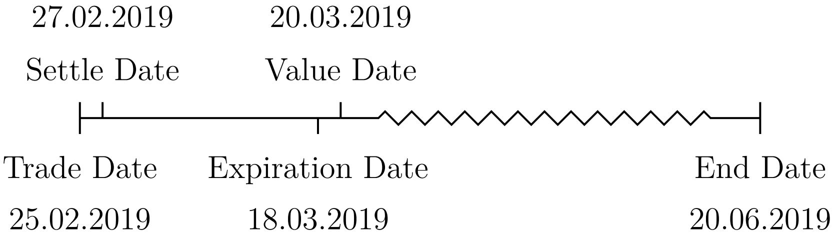
[3] IMM dates, or International Monetary Market dates, are the third Wednesday in months March, June, September and December.

[4] For Short Sterling futures, the expiry will be on the third Wednesday, i.e. no business day lag.

[5] Notional for STIR-futures is equal to the number of contracts multiplied by the contract amount. Contract amount for Eurodollar futures is $\$1,000,000$, for Euribor-futures this is €$1,000,000$, and for Short Sterling futures it is £$500,000$

If the dates for which we cash settle the futures contract align with the payment schedule of an IRS, we could hedge it perfectly. This explains the motivation to price interest rate swaps of forward curves constructed using STIR-futures.

The date schedules of STIR-futures are important for understanding how these instruments are used to price IRSs. Eurodollar and EURIBOR futures are both settled two business days following the trade date. For Short Sterling futures, the settle date coincide with the trade date. All STIR-futures have a cash settlement on the value date, which is the nearest IMM date following the settle date. Eurodollar and EURIBOR futures cash settle against an IBOR prevailing two business days preceding the value date, on the date called expiration date, whereas the Short Sterling futures contract cash settle against an IBOR prevailing on the value date. The IBORs that the contracts are cash settled against start on the value date and end on the end date. A visual example of these five important dates are provided below for an Eurodollar contract bought on February 25th, 2019.



3.1.4 Overnight Indexed Swaps (OIS)

An overnight index swap is similar to an interest rate swap. In contrast to the floating payment in an interest rate swap that is based on an index such as three-month IBOR, the floating payment in an OIS is based on a daily compounding of daily OIS rates over the accrual period of the floating leg.

The underlying index of an OIS is typically rates for overnight uncollateralized lending between banks. Examples of this include the Federal Funds rate in the US, EONIA in the EU and SONIA in the UK. Being an overnight rate between large banks, it captures the low credit risk in a collateralized swap better than IBOR. As a result, both practitioners and academics tend to use OIS rates for discounting payments in interest rate swaps.

Prior to 2007 practitioners used IBOR as a proxy for risk-free rate when valuing derivatives. After the credit crisis that began in 2007, this practice was called into question (Hull & White, 2013). In the wake of the crisis, it has become common to use an OIS rate to discount future cash flows. Hull and White conclude that OIS should be used regardless of the portfolio being collateralized (Hull & White, 2013). In practice, the OIS rate is usually lower than IBOR, meaning that ceteris paribus the swap rates with OIS discounting should be larger than swap rates with IBOR discounting. Additionally, the impact of using OIS rather than IBOR becomes more important with longer IRS tenors.

3.2 Pricing of Interest Rate Swaps

3.2.1 Constructing the Replication Portfolio

Using FRAs and STIR-futures, we show how floating payments in IRSs can be replicated. In particular, we explain how uncertain floating payments can be fixed in advance. Later, we use these replication portfolios to price interest rate swaps.

Consider a trader who is a fixed rate receiver on a one-year USD interest rate swap. The trade date is set to December 18th, 2017 and settled two business days later on December 20th, 2017. This trader needs to pay four floating payments during the life of the swap. The first payment will take place on March 20th, 2018 where the floating rate is fixed to the three-month LIBOR prevailing three months in advance (i.e. trade date December 18th, settle date December 20th, and end date March 20th). The table below reports the realized USD LIBOR rates.

Payment Date	Fixing Date	LIBOR Rates
20.03.2018	18.12.2017	1.63%
20.06.2018	16.03.2018	2.20%
20.09.2018	18.06.2018	2.32%
20.12.2018	18.09.2018	2.34%

These LIBOR rates are only known at the time of their fixing dates. So when the swap is traded, the fixed receiver will only know the LIBOR-rate for the first payment in March, 1.63%. For the second, third and fourth payment, the trader doesn't know what the floating rate will be.

Since December 20th, 2017 is an IMM-date, the value dates for the four next

Eurodollar contracts align almost perfectly with the payment schedule of the interest rate swap. In the table below, we see that the December Eurodollar contract, EDZ17, settles on December 20th, 2017 against a three-month LIBOR rate running from December 20th to March 20th. This is the exact same rate that the first floating payment in the swap is fixed to. However, we must notice that there is a slight mismatch between some of the LIBOR rates the Eurodollar contracts will settle against and the LIBOR rates in the swap payment schedule. As an example, the March Eurodollar contract settles against a LIBOR rate going from March 21st to June 21st, whereas the swap is fixed against a LIBOR rate going from March 20th to June 20th.

Eurodollar		**LIBOR 3M**	
Contract	*Value Date*	*Start Date*	*End Date*
EDZ17	20.12.2017	20.12.2017	20.03.2018
EDH18	21.03.2018	21.03.2018	21.06.2018
EDM18	20.06.2018	20.06.2018	20.09.2018
EDU18	19.09.2018	19.09.2018	19.12.2018

If we ignore these small date mismatches, we can continue constructing the replication portfolio. Fixing the second floating payment in June to a rate known today, involves buying the March Eurodollar contract (EDH18). This contract is cash-settled on March 20th, 2018 against the three-month LIBOR rate, L_{Mar}, prevailing on March 16th, 2018. The value of the cash settlement is given by the difference between the three-month LIBOR, L_{Mar}, and the futures rate, r_{EDH18}, from the March Eurodollar contract bought December 18th, 2017.

This cash settlement will occur on March 20th (i.e. ignoring the one-day date mismatch above), which is three months before the third floating payment. If the payoff is positive, we invest the payoff for three months at L_{Mar}. In three months, on June 20th, we receive the payoff from the futures contract plus the interest from the three month deposit at the same time as we are supposed to pay the floating payment in the swap. Once we sum these payments together, we can see from the table[6] below that we end up receiving the fixed rate, R, and paying the floating leg of the futures rate, r_{EDH18} plus a small adjustment term, $L_{Mar}(L_{Mar} - r_{EDH18})$, that is positive if the payoff from the long position in Eurodollar futures is positive.

[6] Notional, N, and the year-fractions for the accrual periods, τ, are omitted from the payoff table for simplicity.

Table 1: Replication Portfolio if Value of Long Eurodollar Position >0

	March 20th, 2018	**June 20th, 2018**
Eurodollar: Long Position	$L_{Mar} - r_{EDH18}$	0
Invest ED Long @ LIBOR	$-\left(L_{Mar} - r_{EDH18}\right)$	$\left(L_{Mar} - r_{EDH18}\right)\left(1 + L_{Mar}\right)$
IRS: Receive Fixed, Pay Floating	0	$R - L_{Mar}$
Payoff	0	$R - r_{EDH18} + L_{Mar}(L_{Mar} - r_{EDH18})$

This example shows that we are able to set the future payment in the swap to a value we know today, even if we do not know the value for the LIBOR rate that the future floating payment will be fixed at. We removed uncertainty, and as a result are able to hedge the payment almost perfectly today. A similar argument can be made if the value of the long Eurodollar position is negative at settlement. In that case, the trader would borrow at the repo rate instead of investing at LIBOR.

To provide some evidence for this near-arbitrage strategy, we go through a quick numerical example where we use market data. We already know the LIBORs that the floating payments were fixed to. In addition, we have the following data on Eurodollar contracts from December 18th, 2017

Instrument	**Yield**
EDZ17	1.63%
EDH18	1.78%
EDM18	1.92%
EDU18	2.02%

The swap rate for the one-year USD IRS trading on December 18th, 2017 was 1.87%. Assuming quarterly floating payments and semiannual fixed payments, the payment in June will be $1.87\% * 0.5 - L_{Mar} * 0.25$. Using the March Eurodollar contract, the trader would expect a payment of $1.87\% * 0.5 - 1.78\% * 0.25 = 0.4891\%$ in June. Below we see that the actual payment ended up being 0.4897%, which is only 0.06 bps from the the rate the trader fixed in advance, effectively showing that we were able to fix the future cash flow of the swap to a value known today.

	March 20th, 2018	**June 20th, 2018**
Eurodollar: Long Position	0.1067%	0
Invest ED Long @ LIBOR	−0.1067%	0.1073%
IRS: Receive Fixed, Pay Floating	0	0.3825%
Payoff	0	0.4897%

Constructing a similar replication portfolio with FRAs is easier, and we should in theory not experience any discrepancies between the implied forward rates from an FRA and the rate that we are able to lock in.

Table 2: Replication Portfolio if Value of Long FRA Position >0

	March 20th, 2018	**June 20th, 2018**
FRA: Long Position	$\left(L_{Mar} - r^{FRA}_{Mar}\right)/(1 + L_{Mar})$	0
Invest FRA Long @ LIBOR	$-\left(L_{Mar} - r^{FRA}_{Mar}\right)/(1 + L_{Mar})$	$(L_{Mar} - r^{FRA}_{Mar})$
IRS: Receive Fixed, Pay Floating	0	$R - L_{Mar}$
Payoff	0	$R - r^{FRA}_{Mar}$

Motivated by these replication portfolios, we later use STIR-futures and FRAs to replicate the floating legs of interest rate swaps. To do so, there are a number of technicalities that we have not touched upon yet. Later, we see how we can extend the futures-framework to account for non-IMM dates and we also treat the alignment between the payment schedule of the swap and the value dates of the STIR-futures with more care. But first, let us clearly define the pricing formulas used to price interest rate swaps.

3.2.2 General Formula for Pricing Vanilla Interest Rate Swaps

A vanilla interest rate swap has one leg of floating payments and one leg of fixed payments. These payments occur at different frequencies. In this example, we assume the swap in question is a USD interest rate swap with semiannual fixed payments and quarterly floating payments that are fixed to three-month LIBOR three months in advance. The framework below is flexible, and other payment frequencies and conventions can easily be added. At the trade date, swaps are usually priced at par, meaning that the present values of both legs are equal.

The present value of the cash flow from the semiannual fixed leg is based on an annual swap rate, R, multiplied by the correct year-fractions, τ_j^{fix}, to make it reflect the semiannual payment and then discounted back to a present value by multiplying with discount factors, $Z(t, T_j)$

$$PV_{fix}(t) = NR \sum_{j=1}^{m} \tau_j^{fix} Z(t, T_j)$$

where N is the notional of the swap and j would occur at the frequency of the fixed leg (i.e. every six months for USD). The present value of the floating leg is be similar, but the swap rate, R, is replaced by a floating rate, r_i that is fixed to the three-month LIBOR prevailing three months before the floating payment

$$PV_{flt}(t) = N\sum_{i=1}^{n} r_i\tau_i^{flt} Z(t, T_i)$$

where i would occur at the frequency of the floating leg (i.e. every three months for USD).

An interest rate swap priced at par (i.e. $PV_{fix}(t) = PV_{flt}(t)$) should have a present value of fixed payments that is equal to the present value of floating payments. From a fixed-payer perspective, the present value of an interest rate swap can be written as

$$PV(t) = N\left[PV_{flt}(t) - PV_{fix}(t)\right]$$

Solving for the swap rate, we obtain

$$R = \frac{\sum_{i=1}^{n} r_i\tau_i^{flt} Z(t, T_i)}{\sum_{j=1}^{m} \tau_j^{fix} Z(t, T_j)}$$

where $r_i\tau_i^{flt}$ can also be written as $\left(\frac{Z(t,T_{i-1})}{Z(t,T_i)} - 1\right)$

Prior to the 2007-2008 financial crisis, it was commonly accepted to extract both forward rates, r_i, and discount factors, $Z(t, T)$, from the same instruments, for example LIBOR-indexed STIR-futures.

During the financial crisis, the perception of credit risk changed. The idea of IBORs as good proxies for risk-free rates took a hit, especially at the collapse of Lehman Brothers. In the aftermath of the crisis it is common to discount collateralized payments such as the payments in a vanilla USD interest rate swap by OIS rates(Hull & White, 2013). In practice, this involves constructing multiple curves; the forward curve based on IBORs and the discounting curve based on OIS-rates(Veronesi, 2016).

Another perspective introduced during the financial crisis involves payment frequencies of rates with the same tenor. Along with the realization that "risk-free" institutions such as banks could go bust, liquidity started trading at a premium. For a contract of the same length, for example six months, a succession of smaller length contracts, such as two three-month rates (0x3 and 3x6) were deemed less risky than one six-month rate (0x6). Prior to this realization, it would have been acceptable to bootstrap a 3x6 forward rate by combining a 0x3 and 0x6 rate.

When stripping the floating rates, r_i, we need to extract them from a curve that has the same underlying index (White, 2012; Ametrano & Bianchetti, 2013). For a USD swap indexed against three-month IBORs this involves only using instruments with the same index.

Taking the tenor- and discounting-consideration into practice, the swap rate, R, for a USD vanilla interest rate swap can be expressed as

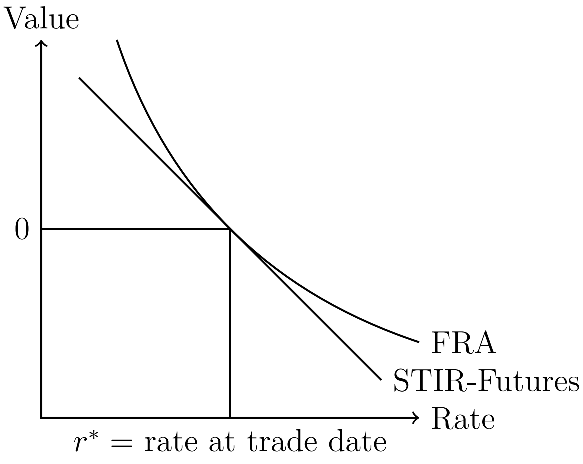
$$R = \frac{\sum_{i=1}^{N} r_i^{IBOR,3M} \tau_i^{flt} Z^{OIS}(t, T_i)}{\sum_{j=1}^{M} \tau_j^{fix} Z^{OIS}(t, T_j)}$$

where $r_i^{IBOR,3M} \tau_i^{flt}$ can also be written as $\left(\frac{Z^{IBOR,3M}(t,T_{i-1})}{Z^{IBOR,3M}(t,T_i)} - 1 \right)$

Thus, finding the swap rate is a matter of constructing a discount curve from OIS rates, and a forward curve from instruments with an underlying index of three-months LIBOR, for example Eurodollar futures. For another currency, for example EUR and GBP, the setup would be adapted to local conventions.

4 Convexity Adjustments for Futures Rates

Implied forward rates from STIR-futures and forward rate agreements are not directly comparable. Consider a trader going short a STIR-futures contract and long an FRA. At the trade date, the value of both legs are equal and the net payoff is zero. However, as the interest rate changes, differences in how the two instruments are settled leads to different valuations. The P&L for the futures contract is settled daily against the variation margin, whereas the P&L for the FRA is only realized once.



The figure above illustrates that the value of STIR-futures move linearly with rate changes. The value of the FRA moves non-linearly, and we see that the value of the futures contract is always lower than or equal to the value of the forward rate agreement contract. To reconcile the differences, it is common to quote the FRA with a lower rate at the trade date, effectively removing the advantage that follows from the non-linear payoff.

This bias, often called convexity bias, becomes more important as the time to contract expiration increases. Several models have been proposed to deal with the computation of this convexity bias. Below, we walk through two such models: the Vasicek model and the Hull-White one-factor model.

4.1 Vasicek Model

The short rate in the Vasicek model follows an Ornstein-Uhlenbeck process and can be written as

$$dr = \kappa(\mu - r)dt + \sigma dz$$

where μ is the unconditional mean of the short rate, κ is the speed of mean-reversion for the short rate, and σ governs the volatility.

This model was first proposed by Vasicek (1977) and gained popularity due to the interpretability of its parameters and the parsimonious setup. Gupta and Subrahmanyam (2000) and Johannes and Sundaresan (2007) use the Vasicek-model to compute convexity adjustments for STIR-futures. In most cases, this model is used together with other models for computing convexity adjustments, thereby giving the researchers a range of intervals for the convexity adjustments.

We provide a calibration procedure of the Vasicek model in the appendix where we also show how convexity adjustments are computed.

The short-rate process under the Vasicek model is often criticized for being unable to capture term-structure dynamics over time. This is mostly due to the way short rates are treated. Under Vasicek, the short rate is independent of time.

4.2 Hull-White One-Factor Model

An alternative to the Vasicek model that addresses some of the major drawbacks, for example the poor fitting to initial term structures, is the Hull-White one-factor model. The Hull-White one-factor model is given by

$$dr = [\theta(t) - ar(t)]dt + \sigma dz$$

where a and σ are constants describing the mean-reversion and volatility of the process. The instantaneous interest rate at time t is defined as $r(t)$, $\theta(t)$ is a parameter that captures the initial term structure, and dz is a Wiener process.

Calibration of the Hull-White one-factor model starts with finding values for the constants a and σ. If the output from a calibrated HW-model is to be used for STIR-futures convexity adjustments, it is common to start with caps, floors or swaptions (Hull, 2018). These instruments are usually quoted in implied volatility. In this thesis, our focus is mostly on swaptions, which are options on swaps. It is common to refer to swaptions by expiry*tenor, so a 1x9 swaption would expire in in one year, and be for a nine-year swap. Given a swaption volatility matrix it is common to use swaptions along the diagonal (often called coterminals) to calibrate the Hull-White one-factor model(Henrard, 2009; Hull & White, 2001). For a 9x9 matrix this involves using implied volatility from swaptions with expiry and tenor 1x9, 2x8, 3x7, 4x6, 5x5, 6x4, 7x3, 8x2 and 9x1.

Calibrating both a and σ can lead to unstable results, and it is common to fix the mean-reversion variable when calibrating the Hull-White one-factor model (Gurrieri, Nakabayashi, & Wong, 2009). It can be shown that the impact of the mean-reversion variable is smaller than the impact of the sigma in the computation of convexity bias, especially for shorter tenors. In that regard, it makes sense to fix the mean-reversion if the goal is to compute convexity adjustments for STIR-futures rates. In our experience, fixing the mean-reversion to a value of around 5% tend to yield reasonable and stable results[7].

After the Hull-White one-factor model is calibrated, we can use the parameters to compute convexity adjustments. Hull (2017) shows that the convexity adjustment

[7] Sokol (2014) finds that the mean-reversion for most major currencies historically have been between 3% and 10%.

for STIR-futures can be computed as

$$cvx^{HW} = \frac{B(T_1,T_2)}{T_2 - T_1}\left[B(T_1,T_2)(1-e^{-2aT_1}) + 2aB(0,T_1)^2\right]\frac{\sigma^2}{4a}$$

where $B(t,T) = \frac{1-e^{a(T-t)}}{a}$

To compute the convexity-adjusted futures rate, we must transform futures rates from quarterly compounded to continuously compounded

$$r^{STIR,CC} = ln\left(1 + \frac{r^{STIR}}{4}\right)4$$

before subtracting the convexity adjustment

$$r^{STIR,CC,Cvx} = r^{STIR,CC} - cvx^{HW}$$

and transforming the convexity-adjusted futures rate back to quarterly compounding

$$r^{STIR,Cvx} = \left[exp\left(\frac{r^{STIR,CC,Cvx}}{4}\right) - 1\right]4$$

5 Data and Computation of the Bases

5.1 Raw Data from Bloomberg

We use daily last prices from Bloomberg. The data used is for USD, EUR and GBP. Our main focus in this section is the USD data. However, when there are substantial differences between the currencies, we provide explanations on how we treat those differences.

For the three-month deposit, we use the USD LIBOR three-month rate. Forward rate agreements denominated in USD and with three-month USD LIBOR as underlying are available from 2004 for contracts up to two years (i.e. up to 21x24). The FRAs used for EUR and GBP have six-month IBORs as underlying (i.e. 0x6, 6x12, etc.), and their data start from the early 2000's.

For STIR-futures, we use all 40 Eurodollar futures contracts given that they are available. Our sample starts in 1986 where we have the first eight Eurodollar contracts available. From the end of 1993, last prices on all 40 Eurodollar contracts are available. The liquidity on the short end of the futures curve is generally deep whereas the liquidity of the long end can be poor, especially in during the 1990's and early 2000's. STIR-futures for EUR and GBP are EURIBOR futures and Short Sterling futures, both of which use three-month IBORs as the underlying index. The history of the Short Sterling goes back to the 1980's whereas the EURIBOR futures start in the late 1990's.

In addition to forward rate agreements and Eurodollar contracts, we include swap rates for USD-denominated vanilla interest rate swaps with three-month LIBOR as the underlying index. We use swap rates with tenors 2-, 3-, 5-, 7-, and 10-years. These swap rates have coverage back to 1988. For EUR and GBP, the most common IRSs, and thus most liquid, are IRSs quoted with six-month IBORs as underlying index. To reconcile the EUR- and GBP-swaps with the STIR-futures having three-month IBORs as underlying index, we also gather data on tenor basis swaps where three-month IBORs are exchanged with six-month IBORs. Using these, we are able to convert the IRSs in these two markets from having six-month IBOR as underlying to having three-month IBOR as underlying. This computation is based on arbitrage, and it yields reasonable values for synthetic IRSs with three-month IBOR as underlying index. It is possible to obtain EUR- and GBP-swaps that use three-month IBORs as the underlying index, but the historical sample is smaller and the liquidity is poorer, especially prior to 2008.

There are a number of conventions that are specific to interest rate swaps. In isolation, we could proceed and ignore them in our computations later on. However, when considered together, the number of conventions is large enough to have a substantial impact on the basis. A section in the appendix is dedicated to the data sources used and specific conventions for the IRSs. Table 9 in the appendix presents an overview of the most important conventions, and all variables used are summarized in table 10 in the appendix.

The unsecured overnight rate for the US is the Federal Funds rate, so the OIS rates in this market have the one-day Federal Funds rate as underlying. Although the data go back to 2002 in the US market, the usage of OIS rates as a discounting measure for interest rate swaps did not start increasing until after the 2007/2008. We extract OIS rates on 6M, 9M, 1Y, 18M, 2Y, 3Y, 4Y and 5Y before we compute synthetic OIS rates for 7- and 10-years, consistent with the methodology Bloomberg uses for its construction of OIS curves in its "Swap Curve Builder" on the Bloomberg Terminal. To compute these synthetic OIS rates, we combine

LIBOR swap rates and a LIBOR-Fed Funds basis swaps (Bloomberg Quantitative Analytics, 2017). For EUR and GBP we use OIS rates with EONIA and SONIA as the underlying index.

5.2 Date Schedules for Interest Rate Swaps and STIR-Futures

Allocating a specific section for the construction of date schedules may seem like overkill. In our experience however, these date schedules are far from trivial. They are core components of the interest rate derivatives used and extremely important for the basis computations. In this section, we only cover the date schedules from a USD interest rate swap perspective. The conventions used in an EUR IRS is similar to its USD relative, but the GBP IRS can be quite different.

We used Python to compute the basis. In doing so, we extensively used a library called QuantLib, especially for the construction of date schedules. QuantLib is an open-source software framework for quantitative finance (Ballabio et al., 2019). It is written in C++, but it includes functionality that allow Python and other popular scripting languages such as R to access the library. One good resource for Python-use of the QuantLib library can be found in Ballabio and Balaraman (2017).

One benefit from using QuantLib is its ability to account for currency-specific conventions when constructing date schedules for interest rate swaps. Each currency has a specific calendar with particular holidays. If we run into any of these holidays, interest rate swaps have specific conventions on whether one should go forward or backward one business day if that occurs. Most swaps use what is called "modified fo [illegible] go forward, except if that day is in another month, then we should go backward.

Another detail in the construction of date schedules for interest rate swaps is whether we want to construct the schedule from the settle date to the last payment date, or in the reverse. Most swaps use what is called "backward(EOM)". Despite seeming slightly esoteric at first, this means that for a quarterly payment schedule, we start at the last payment date and then go backwards in three-month intervals until we reach the settle date. On most dates or for shorter tenors, going either way (i.e. starting on settle date or starting on last payment date) cause no differences or leave a small impact on the general date schedule. There are however dates where the difference can be large, and using the QuantLib library to account for this allows us to compute synthetic swap rates whose underlying process is closer to the market swap rate. To further complicate the date schedules, the date schedule for two swaps having the same settle date, but different tenors, could be slightly different. Using the backward-convention, the date schedule of a 2-year swap will on most dates be equal to the first two years in the payment schedule of a 5-year interest rate swap. However, if the last payment date for the 2-year swap does not fall on the same day as the last payment date for the 5-year swap, the date schedules could differ by a few days.

Using a framework such as QuantLib for construction of these date schedules allows the process to be more efficient and less prone to human error. It also enables us to better account for country-specific conventions. In isolation, details like this can be ignored. However, pricing interest rate swaps using STIR-futures involves several examples of these small details. Taken together, they might account for an error of several basis points, enough to make our basis computation less robust.

We assume that the date schedules of the forward rate agreements perfectly align with the computed payment schedules of swaps. This is not entirely true, since it will sometimes yield mismatches of the magnitudes one or two days. However, the effect from this is minor, and FRAs are in general less sensitive to dates than STIR-futures.

The date schedules for STIR-futures are less complex than the dates for the IRSs, but still far from straightforward. On any given date, there are five dates we want to compute for STIR-futures. The first and easiest is the trade date, which is set equal to the current date. Next, the settle date is set two business days following the trade date. Except for the need to find two business days in a specific calendar type, the computation of these two dates are straightforward. Next we find the value date of the futures contracts. This involves finding the next IMM date, or the next third Wednesday in a month in the March quarterly cycle. At this point, we also need to account for the way Bloomberg reports futures prices. As an example, the data reported two business preceding the IMM-date in December would be the prices for a futures contract expiring in December. The data reported one business day preceding the IMM-date in December would be for a futures contract expiring in March. We use a function from QuantLib that finds the next IMM-date given a date input where we also reconcile it with the specific data from Bloomberg around IMM-dates. The two last dates needed for the STIR-futures are easier. The expiration date is set as two business days preceding the value date, and the end date is set as three months after the value date. The end date is the end date of the IBOR-index that the futures contract settles against. This type of methodology is described briefly in Ametrano & Bianchetti (2013) and our date schedules align perfectly with examples provided in their paper.

Once these date schedules are constructed, we want to compute year fractions between payments in order to obtain the year-fraction variables, τ, that we have defined earlier. Although the two dates we want to compute the year-fraction between are the same, the year-fraction will be slightly different depending on what day-count convention we want to use. For USD swaps this is 30/360 for the fixed leg and ACT/360 for the floating leg. Using QuantLib, we are able to compute year-fractions that account for this.

5.3 Computing the Basis

The process for computing the basis for forward rate agreements and futures contracts can be divided in three. First, we compute discount factors from both IBOR- and OIS-rates. Then, we align these discount factors with the date schedule of the interest rate swap by interpolation. Lastly, we use the pricing framework developed earlier in the thesis to compute the futures- and forward-implied swap rates before computing the basis.

For the futures-implied swap rates, we first construct an IBOR-indexed yield curve by combining three-month deposits and STIR-futures. We start by finding the end date of each contract, meaning settle date plus three months for the deposit and the end date for the STIR-futures as defined earlier. We rank the annualized rates, r_i, from shortest time-to-maturity to longest. Using the year-fractions, τ_i, between end dates of the instruments, which is around 0.25 in most cases, we compute discount factors by taking a cumulative product

$$Z^{STIR}(t,T) = \prod_{i=1}^{T} \frac{1}{1 + r_i \tau_i}$$

where t is set equal to the settle date for the deposit- and futures-rates, and τ_i accounts for the specific day-count convention of the instruments, ACT/360 for USD deposits and STIR-futures. If we wanted to do this for forward rate agreements, we would keep the first three-month deposit and then replace r_i with rates from FRA's instead of STIR-futures contracts.

The method for stripping discount factors from OIS-rates differs from the method of their IBOR counterpart. Let OIS_N be the OIS-rate for the Nth maturity and τ_i be the year-fraction between maturity dates of a stream of OIS-rates (0.25 for a quarterly difference in maturity dates). Using a methodology outlined by Smith (2013), discount factors from OIS-rates up to one year can be extracted by

$$Z^{OIS}_{\leq 1Yr}(t, T_N) = \frac{1}{1 + OIS_N \sum_{i=1}^{N} \tau_i}$$

For OIS-rates with tenor longer than one year, we use a recursive procedure to extract discount factors

$$Z^{OIS}_{>1Yr}(t, T_N) = \frac{1 - OIS_N \sum_{i=1}^{N-1} Z^{OIS}(t, T_i)\tau_i}{1 + OIS_N \tau_N}$$

After computing the discount factors, we align them so that they fit with the date schedules of the interest rate swaps. We use IBOR discount factors to find the forward curve for the swap. Given a set of IBOR discount factors, we compute the year-fraction between the settle dates and the end dates of the IBOR-indexed rates. In addition, we have the year-fractions between the settle dates and the payment dates in the interest rate swap date schedule. As we have seen earlier, these dates only align well on IMM-dates.

For perfect alignment, we use log-linear interpolation on the IBOR discount factors to obtain synthetic discount factors that align with the swap date schedule. This involves doing a log-transformation on the discount factors followed by linear interpolation. The choice of this method is mostly motivated by the short distance (i.e. three months) between the IBOR-instruments we are using. In addition, since the liquidity of some STIR-futures contract, especially at the long end of the curve can be poor at times, the local dependency in a linear interpolation scheme is preferred compared to global dependency in cubic interpolation.

Given the cumulative year-fraction, κ_i, for a payment happening at time i, we compute the interpolated discount factor, $Z^{STIR}(t, T_i)$, by using

$$\begin{aligned} log\left[Z^{STIR}(t, T_i)\right] &= log\left[Z^{STIR}(t, T_{i-\Delta})\right] \\ &\quad + \left(log\left[Z^{STIR}(t, T_{i+\Delta})\right] - log\left[Z^{STIR}(t, T_{i-\Delta})\right]\right) \frac{\kappa_i - \kappa_{i-\Delta}}{\kappa_{i+\Delta} - \kappa_{i-\Delta}} \end{aligned}$$

where we find the two nearest known dates on each side of i, denoted by $i - \Delta$

and $i + \Delta$, and use the known discount factors on these dates to interpolate for the discount factor at time i. An example of this can be seen in figure 1 below.

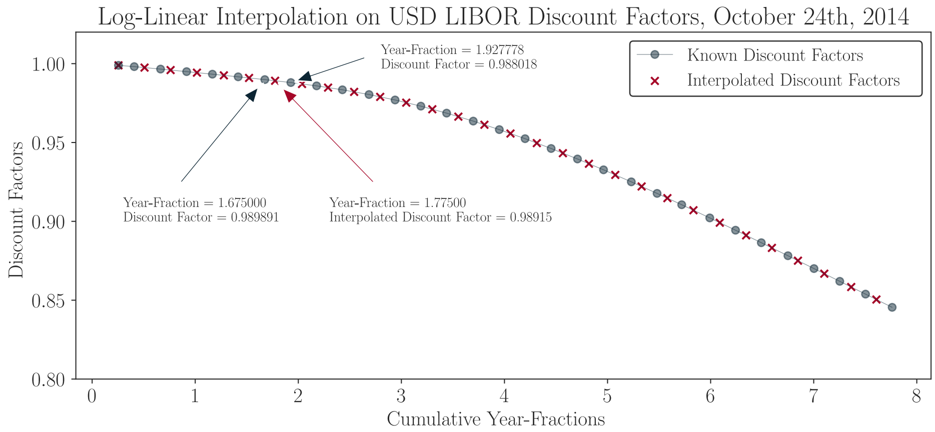


Figure 1: Log-Linear Interpolation for USD LIBOR Discount Factors

For the discounting curve, we use OIS discount factors and interpolate discount factors at swap payment dates by using log-cubic interpolation. This type of interpolation is a common choice when interpolating along the OIS-curve (Ametrano & Bianchetti, 2013; Darbyshire, 2016).

Most popular programming languages have built-in functions that allow users to set up log-cubic interpolation schemes. Ron (2000) provides an overview of technical details regarding its implementations and Darbyshire (2016) highlights a number of practical concerns surrounding its implementation. Intuitively, interpolating discount factors through log-cubic interpolation involves choosing a number of knot points, in this case the known discount factors, for which we fit a cubic polynomial through. Naturally, as the number of knot points increase, the result from log-cubic interpolation becomes more similar to the results from log-linear interpolation. In practice, care should be taken, especially regarding what known discount factors should be used as knots. By including a discount factor for a tenor that is illiquid, one risks obtaining a cubic polynomial that do not fit the data well.

After interpolating for the discount factors, $Z^{OIS}(t,T)$, we compute the futures-implied swap rate, R^{Fut}, by equating the present values of the floating leg and fixed leg

$$R^{Fut} = \frac{\sum_{i=1}^{N} \left(\frac{Z^{STIR}(t,T_{i-1})}{Z^{STIR}(t,T_i)} - 1 \right) Z^{OIS}(t,T_i)}{\sum_{j=1}^{M} \tau_j^{fix} Z^{OIS}(t,T_j)}$$

where $\left(\frac{Z^{STIR}(t,T_{i-1})}{Z^{STIR}(t,T_i)} - 1\right) = r_i^{STIR}\tau_i^{flt}$

We compute futures-implied swap rates for 2-year, 3-year, 5-year, 7-year and 10-year tenors in all markets given that we have enough STIR-futures to cover all tenors.

Through this setup we are able to respect both the tenor- and discounting consideration that we touched upon earlier. By using OIS-rates to discount the cash flows we are implicitly assuming default risk close to zero. This is a reasonable assumption for collateralized swaps, and in line with what Johannes and Sundaresan (2007) who claimed that interest rate swaps should be discounted at a rate lower than LIBOR. This type of discounting has become market practice, and the Bloomberg data is discounted with OIS. Despite this, we did not experience that the choice between IBOR- or OIS-discounting made a large impact on the futures-implied swap rates for most time-periods in our sample. An example of this can be seen in the figure 1 below.

To compute the futures basis, we subtract the market swap rate, R, from the futures-implied swap rate

$$Basis^{Fut} = R^{Fut} - R$$

For the swap rates used to compute the futures basis for EUR and GBP, we use tenor basis swaps to convert the six-month underlying to a three-month underlying. This is easiest to illustrate with GBP swaps. The GBP tenor basis swap have one leg with the six-month GBP LIBOR and one leg with the three-month GBP LIBOR. Both legs have the day count convention ACT/365. The payment frequency for the six-month leg is semiannual and the payment frequency for the three-month leg is quarterly. The GBP IRS has semiannual payments for both the floating and fixed leg, the day count conventions are ACT/365, and the floating payment is based on a six-month GBP LIBOR. Using tenor basis swaps, we have

	0.25	**0.50**	**0.75**	**1.00**
GBP IRS: Receive Fixed, Pay Floating		$R - r^{6M}$		$R - r^{6M}$
GBP Tenor Basis Swap: Receive 6M, Pay 3M	$-r^{3M}$	$r^{6M} - r^{3M}$	$-r^{3M}$	$r^{6M} - r^{3M}$
Payoff	$-r^{3M}$	$R - r^{3M}$	$-r^{3M}$	$R - r^{3M}$

created a new one-year GBP interest rate swap with quarterly floating payments that is reset against a three-month IBOR, and the fixed payments are as before. This allows us to use STIR-futures directly for the computation of the basis.

The method for implied swap rates using forward rate agreements is similar to the one presented above with the main difference meaning that we do not need to interpolate since we assume that the end dates align perfectly well with the date schedule of the swap. In addition, there is no need to include tenor basis swaps for EUR and GBP since the FRAs we use are indexed to six-month IBORs.

5.4 Computing the Convexity Bias

One popular model for computing convexity bias on STIR-futures is the Hull-White one-factor model. It fits within the no-arbitrage family of term-structure models, and is used both empirically and in practice[8].

We also tried computing convexity bias with a Vasicek model. In our experience, the dynamics of this model failed to capture the behavior of the interest rates, and as a result the convexity adjustments were more or less constant over time. This lead us to change the model used to the Hull-White one-factor model.

The computation of the convexity bias under the Hull-White one-factor model involves two parts: the calibration of the one-factor model where we obtain the two parameters for mean-reversion and sigma, and the part where we use those parameters to compute the convexity bias. Of the two parts, the calibration is the most technically challenging.

For all currencies, we use European-style at-the-money coterminal swaptions whose tenor+expiry is equal to ten (i.e. 1x9, 2x8, etc.). In the US, where the coverage is best, we use both options quoted in lognormal volatilities from 2000, and normal volatilities from 2005/2006. For EUR Swaptions, we only have normal volatilities available and the coverage runs from 2006 to 2019.

Intuitively, the calibration process involves minimizing the difference between market prices of swaptions and model prices of swaptions, and then use those parameters in the Hull-White one-factor model. We implement this calibration in Python using the QuantLib library. Using built-in functions, we are able to efficiently calibrate the Hull-White model for both USD and EUR swaptions. Good resources for implementing the Hull-White model in Python can be found in Ballabio and Balaraman (2017) and Katajamäki (2017).

[8] Convexity adjustments on the Bloomberg terminal are based on the Hull-White one-factor model.

We fix the mean-reversion parameter to 5%, consistent with previous theory, and calibrate the volatility parameter by minimizing the difference between market prices and model prices of coterminal swaptions. After calibrating the mean-reversion and volatility, we use the parameters to compute convexity adjustments for STIR-futures. To do so, we follow a setup described in Hull (2017) using formulas outlined earlier in the theory section.

Both parameters obtained for the Hull-White model and the resulting convexity bias have been checked with sources such as Ametrano & Bianchetti (2013) and it yields similar convextiy adjustments as Burghardt(2003)[9]. In addition, a simple plot of the level of convexity bias shows logical results. The bias is larger for futures-contracts with longer tenors, and it is higher during periods of increased market stress.

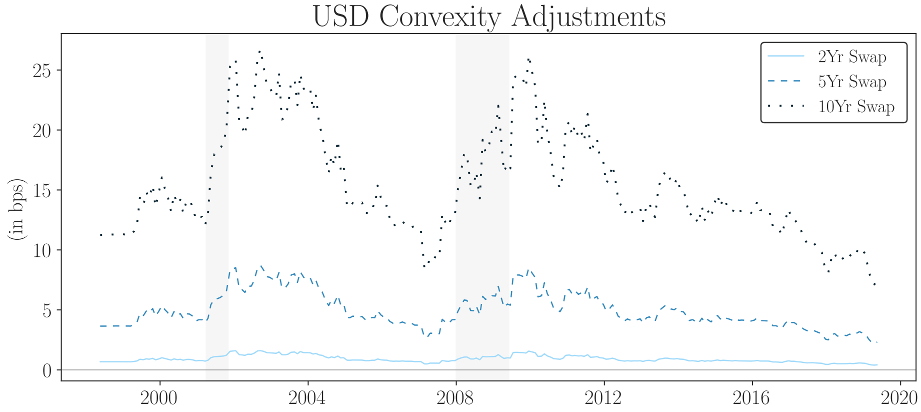


Figure 2: Convexity Adjustments for USD Swap Rates Using Hull-White One-Factor Model

6 Implied Swap Rates

Following the methodology outlined above, we compute bases for currencies USD, EUR and GBP using both STIR-futures and forward rate agreements. The level of market swap rates in these currencies is given in figure 3. Previous research (Gupta & Subrahmanyam, 2000; Minton, 1997; Johannes & Sundaresan, 2007; Burghardt, 2003; Bomfim, 2003) and economic rationale suggest that the magnitude of the bases should be close to zero for short tenors such as two years,

[9] Burghardt did not use Hull-White one-factor model for his convexity bias, but the results should nevertheless be similar, something which they are.

and somewhat larger, but not larger than 10-15 bps for longer tenors such as five years. The size and liquidity of these markets mean that they are largely driven by traditional supply- and demand-mechanics so large mispricings should disappear quickly. We find similar evidence. Surprisingly, we also find large and persistent basis, especially using GBP-denominated FRA's in the period after the referendum for Brexit was announced in 2016, suggesting that frictions do exist.

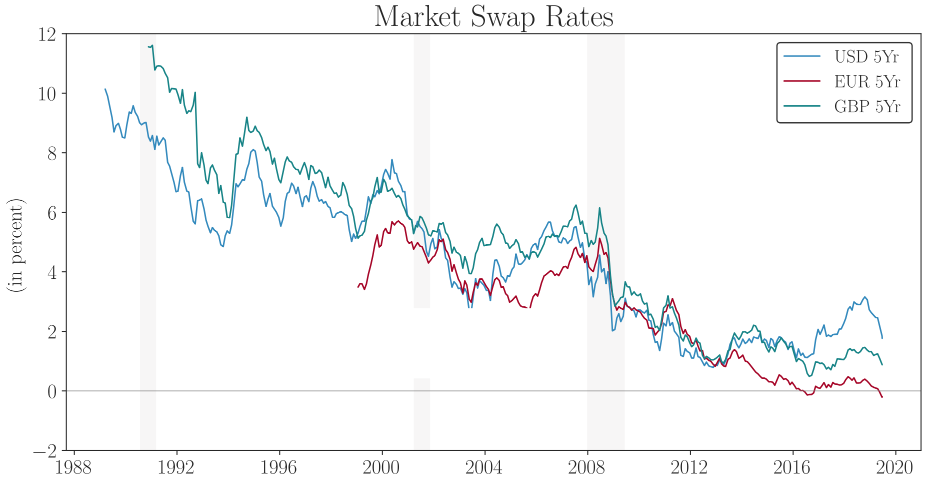


Figure 3: 5- ates in USD, EUR and GBP

Burghardt (2003) provides examples of the economic impact of the basis. A 5-year swap mispriced by 2-3 bps is worth around $\$80,000$ on a 100 million IRS. For swaps of longer tenors, the impact is larger.

In the next sections, we present descriptive statistics on the magnitude and behavior of the basis in the three currencies using different instruments. We split the data in three periods. Pre-crisis is before 2007, crisis is from 2007 to 2009, and post-crisis is after 2009. The figures below are all sampled at a monthly frequency, more specifically the third Wednesday in a month. This choice is motivated by the replication portfolio described earlier so that we are able to capture the basis on some IMM-dates. All tables are based on daily data.

All bases are cross-referenced with available sources to verify their correctness. For the date schedule for the futures rates, we compare with Ametrano and Bianchetti (2013). For the date schedules on the interest rate swaps we use Labuszewski (2010) and Burghardt (2003). Most previous research work with other types of interest rate swaps. For example, the USD IRS used to compute the futures basis in Gupta and Subrahmanyam (2000) has a six-month LIBOR as the underlying index and some different conventions compared to the swap here. Burghardt (2003) provides an example in his section on the convexity bias, and when comparing with the Eurodollar-implied swap rates he obtains, our futures-implied swap rates are similar, leading us to believe that our computations are correct.

Table 3: Comparison with Futures-Implied Swap Rates from Burghardt, September 10th, 2002

	Without Convexity					With Convexity				
	2Yr	**3Yr**	**5Yr**	**7Yr**	**10Yr**	**2Yr**	**3Yr**	**5Yr**	**7Yr**	**10Yr**
Author Rates	2.4988%	2.9858%	3.7121%	4.2460%	4.7862%	2.4831%	2.9520%	3.6271%	4.0962%	4.5265%
Burghardt Rates	2.5040%	2.9890%	3.7180%	4.2600%	4.8040%	2.4900%	2.9550%	3.6300%	4.1060%	4.5360%
Difference (in bps)	0.5186	0.3201	0.5926	1.3997	1.7841	0.6888	0.2996	0.2890	0.9843	0.9519

Note that these rates are computed using IBOR discounting. For dates when OIS-data is available, we also compute implied swap rates with OIS discounting, although it does not impact the futures basis much as seen in figure 8 in the appendix.

6.1 Forward Basis

Empirical works have mostly focused on theoretical forward rates instead of using market prices of FRAs. This is mostly due to lack of data. Using data from Bloomberg, we have FRAs in USD from 2004, in EUR from 2000 and in GBP from 2000.

Johannes and Sundaresan (2007) compute theoretical futures- and forward-rates using term-structure models such as Vasicek and Cox, Ingersoll and Ross. They posit that the market swap rate should lie above the swap rate implied by forward rates, and below the swap rate implied by futures rates. If this holds, we would expect to see a negative forward basis.

Table 5 provides summary statistics on the forward basis in the currencies USD, EUR and GBP. Both the mean and median for the forward basis in all currencies are closer to zero in the post-crisis period relative to the pre-crisis period, consistent with more trading and better liquidity in the market for FRAs. The bases for USD and EUR both fluctuate around zero, and their means are negative in all the three sub-periods we defined above, seemingly following the theory outlined by Johannes and Sundaresan (2007). The basis for GBP has a positive mean and it's magnitude is a few basis points, ranging from 3.7 bps in the post-crisis period to 5.8 bps in the pre-crisis period. Looking at figure 4, it is clear that the GBP forward basis experienced longer periods of a positive basis in the pre-crisis period, whereas the positive periods have been centered around shorter time-intervals in the post-crisis period.

The standard deviations of the forward bases are all higher during the crisis-period with the span of values taken on by both the USD and GBP bases being wider compared to the two other periods. Standard deviations returned to pre-crisis levels for both USD and EUR in the post-crisis period, whereas the GBP standard deviation has remained high, potentially due to specific events such as Brexit, a term used for describing the process following United Kingdom's decision to leave the European Union[10].

As is evident in figure 4, the basis is not as close to zero as our replication portfolio argument suggests, especially for currencies USD and GBP. The USD basis does not exhibit persistent positive or negative levels, making it difficult to attribute the basis to one particular driver. The GBP basis is large and persistent, particularly between 2016 and 2018.

6.2 Futures Basis

The futures basis has been treated extensively in previous literature. Minton (1997), Gupta and Subrahmanyam (2000), Bomfim (2003) and Johannes and Sundaresan (2007) all examine the basis and provide insights on its behavior. Most theory finds that the basis should be positive and for shorter tenors such as two years it should be close to zero. This is similar to what we find in all currencies.

Table 6 provides descriptive statistics on the futures basis in currencies USD, EUR and GBP. The mean for the USD and EUR bases are positive in all three sub-periods. The mean for the GBP basis is negative in two sub-periods, but it is -0.07 and -0.08 bps meaning that it can be thought of as zero. The mean of the bases are all close to zero in the three sub-periods.

Standard deviations of the bases exhibit similar levels in the pre- and post-crisis periods. Unsurprisingly, the standard deviations increase during the crisis-period, in particular for the USD futures basis, which makes sense since this was the center of the credit crisis of 2007/2008.

[10] Referendum announced in February 2016, and referendum held in June 2016.

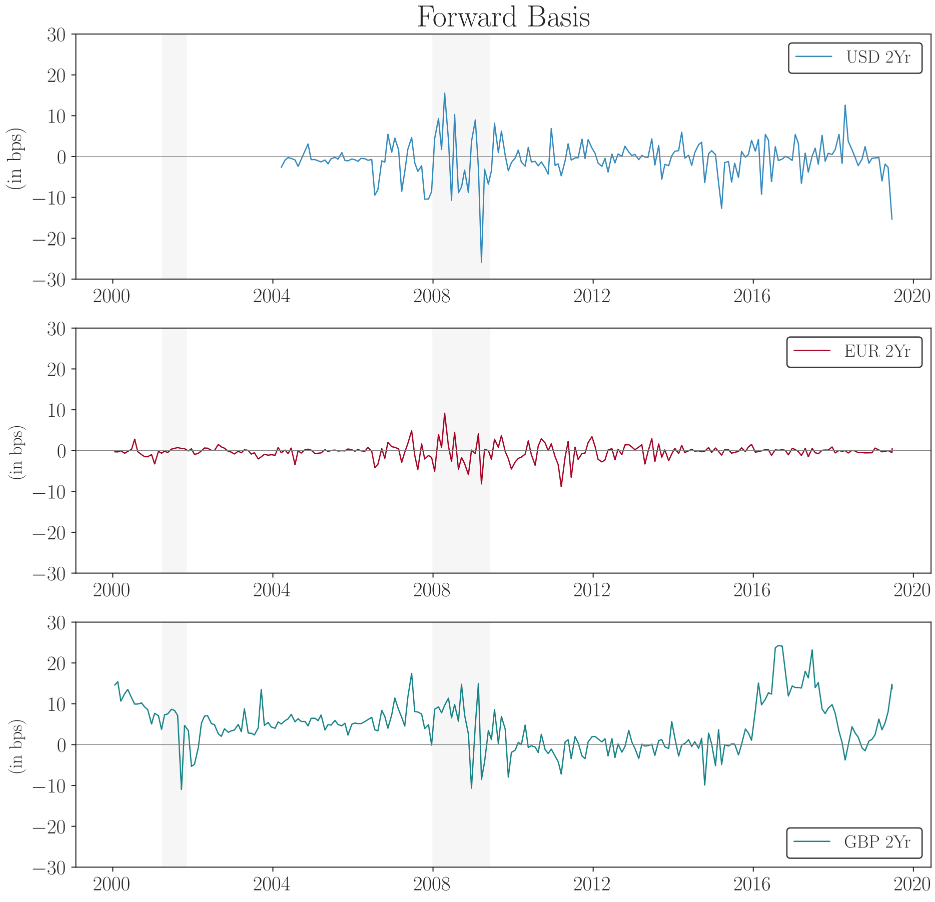


Figure 4: Forward Basis for 2-Year in USD, EUR and GBP

The mean bases for the USD and GBP bases increased from the pre-crisis period to the post-crisis period. The basis in EUR did not change much and it is currently the basis closest to zero. The EUR interest rate swap has the lowest rates throughout this time-period and it is also the futures basis closest to zero. The futures basis for the 2- and 5-year tenors are both close to zero suggesting that there were few exploitable arbitrage opportunities in this currency during our historical sample. In comparison, around 50% of the daily two-year USD futures basis observations have an absolute value larger than 3 bps.

Figure 5 plots the 2- and 5-year futures basis for currencies USD, EUR and GBP. The 2-year USD futures basis is fluctuating around zero, with few long periods in positive or negative territory. The 5-year basis tends to be more on the positive side of the basis, and it is also larger than the 2-year basis. This is consistent with the effects reported by Gupta and Subrahmanyam (2000) that convexity is one of the main drivers of the basis, and that it should be larger for longer tenors. An example of this can be seen in figure 8 in the appendix where we show the effect of convexity adjustment for the seven-year USD futures basis.

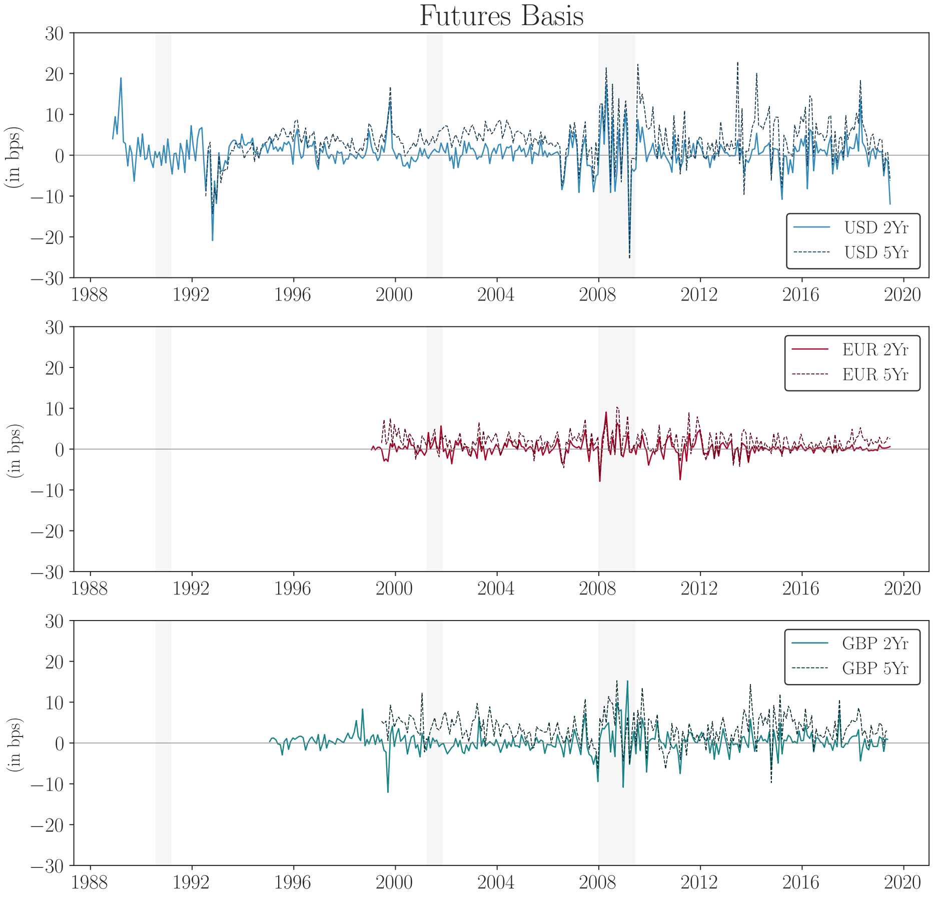


Figure 5: Futures Basis for 2- and 5-Year in USD, EUR and GBP

6.3 Basis Across Instruments

In this section, we look closer at the correlations between the futures- and forward bases in individual currencies. Economic rationale posits that the correlations should be similar unless instrument-specific risk factors are present. Examples of this include credit risk in an FRA versus the more collateralized futures contract, or the liquidity concerns related to posting an initial margin and keeping up the variation margin in a futures contract relative to an FRA with one payment only.

Figure 6 provides a scatter plot of the 2-year futures- and forward-basis in the currencies USD, EUR and GBP. The x-axes show the level of the forward basis, and the y-axes show the level of the futures basis. The scatter plots are divided in three time-periods that we are familiar with from before.

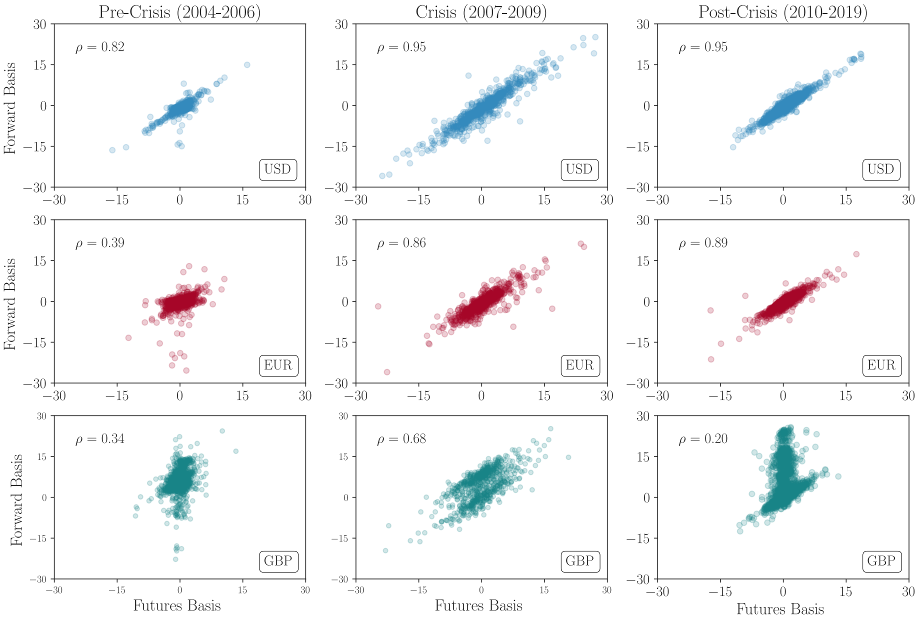


Figure 6: Correlation Between Futures- and Forward-Basis

The correlation[11] between the USD futures-and forward basis is strong at 0.82 in the pre-crisis period, and 0.95 in the two other periods. This is less surprising in the pre- and post-crisis periods where we would expect the two bases to move similarly. However, during the crisis, when credit risk became more important, it is noteworthy that market participants perception of credit risk did not reduce the correlation between the collateralized futures-contracts and uncollateralized forward rate agreements.

A similarly strong correlation during the crisis-period compared with other periods can be witnessed in the two other markets as well.

Neither the EUR nor GBP bases displayed strong correlation in the pre-crisis period. From the scatter plot in figure 6 we can see that this is mostly driven by

[11] Pearson correlation coefficient.

occasional large values in the forward basis, potentially driven by less trading in these types of products in the pre-crisis period.

In the post-crisis period, the EUR bases display strong correlation. Both EUR bases have lower absolute levels compared to the other markets during this period. The correlation between the GBP bases display weak correlation in the post-crisis period. This is more visible from figures 4 and 5 where we saw that the GBP forward basis reacted more to the Brexit event compared to the GBP futures basis.

6.4 Basis Across Markets

After looking at the correlation between instruments, we turn our focus to correlation across markets. Economic rationale asserts that correlations between markets should increase during global financial crises. Additionally, a negative driver of the correlation could be country-specific regulations or actions by central banks.

Figure 9 in the appendix shows the 2-year bases correlation between the forward bases in different currencies in the pre-crisis-, crisis-, and post-crisis-periods. Correlations across markets were low in the pre-crisis period, ranging from 0.13 for GBP/USD to 0.21 for EUR/GBP and 0.46 for EUR/USD. During the global financial crisis of 2007/2008, correlations increase which in line with assertions from theory. The EUR/USD correlation grows to 0.60 and the correlation between EUR and GBP grows to 0.46. These correlation-levels are not strong, but they nevertheless exhibit a linear relationship between the currencies, which becomes evident as we look at the scatter plots in the appendix.

In the post-crisis period, correlations between markets revert back to pre-crisis levels. The correlation between USD and EUR is of particular interest. This correlation moves from 0.46 pre-crisis to 0.60 during the crisis to 0.28 in the post-crisis period. One potential explanation for this collapse could be specific regulations or actions by the central banks of each currency. As we have touched upon earlier, the EUR forward basis is trading at levels closer to zero compared to the other markets.

A similar figure for the futures bases is presented in figure 10 in the appendix. At a general level, the correlations are similar to what we observed for the forward bases. Compared to the correlations between the forward bases, the correlations between the futures bases do not increase as markedly during the crisis. It is also noticeable how correlations revert back to pre-crisis levels in the post-crisis period. The only exception, which is also similar to what seen in the forward bases, is the correlation between USD and EUR, which drops from 0.52 in the pre-crisis to 0.25 in the post-crisis further suggesting that something happened in the EUR market that separated it further from the USD market.

7 Limits to Arbitrage

We turn our focus towards factors explaining why apparent arbitrage opportunities are not exploited by arbitrageurs. In particular, we look at the GBP forward basis between 2016 and 2019 and the USD futures basis for longer tenors. We start by introducing factors related to credit risk and liquidity. After introducing their intuition and linking them with the specific arbitrages, we propose some specific factors for each arbitrage strategy.

Credit risk can be defined as the risk of partial or complete loss of future cash flows when a counterparty fails to honor the contract. More concisely, credit risk is the risk of default. Previous research asserts that posting of collateral close to eliminates credit risk (Collin-Dufresne & Solnik, 2001; Liu et al., 2006; Feldhütter & Lando, 2008). As a result, an uncollateralized FRA is likely to include more credit risk than futures contracts or interest rate swaps.

Credit risk can be measured in multiple ways. The TED spread was used by Gupta and Subrahmanyam (2000) for measuring credit risk. It is defined as the spread between a three-month IBOR and Treasury bills of the same maturity. An alternative might be the IBOR-OIS spread. This spread shows the risk premium for an interbank rate with implicit credit risk (i.e. the IBOR) relative to an unsecured overnight rate with no credit risk (i.e. a "risk-free" rate). In times of market stress, when uncertainties around counterparties are larger, the IBOR-OIS spread increases, reflecting the increased premium of interbank lending.

Liquidity is another well-studied factor in previous literature. In times when liquidity is easily accessible, the replication portfolio described earlier should be easier to trade, thereby reducing any near-arbitrage profits. In such times, an arbitrageur could easily enter into positions large enough to benefit from the basis. We have seen that the bases are of a smaller magnitude, so fully capturing economic profits require a generous amount of leverage. To test the effect of liquidity on the arbitrage-strategies, we use tenor basis swaps. The swap rate of a 1M vs. 3M tenor basis swap represents the premium paid to receive a succession of three one-month rates instead of one three-month rate. At the payment dates in the swap, a one-month rate is swapped against a three-month rate, insofar the payment dates of the two legs align. The rates swapped are all indexed against the local IBOR.

We echo the interpretation of Du, Tepper and Verdelhan (2018) and use the tenor basis swap as a measure of liquidity rather than credit risk. Darbyshire (2016) extends the notion to also include optionality. Optionality in this case is best understood through an example. An investor lending at a three-month rate locks itself to one counterparty throughout the entire three-month period, creating a rather illiquid asset. An investor lending three consecutive one-month rates has the option of reallocating the second and third one-month lending period to other counterparties or other endeavours, wherever the expected return is higher. This built-in option has more value in times when uncertainty and liquidity is of higher importance. Thus, liquidity and optionality become related.

7.1 The GBP Forward Basis Arbitrage

The FRA replication portfolio postulates that the forward basis should be zero. Despite this, the GBP forward basis traded between 10 bps and 20 bps for longer periods between 2016 and 2019. Compare that to levels around 0 bps in EUR, and between -5 bps and 5 bps in USD, and it seems clear that there should be a theoretical arbitrage. An arbitrageur would short the FRAs and go long a fixed receiver IRS, locking in a profit of 10-20 bps multiplied by the notional and the year-fraction for the fixed payments during the two-year swap. This deviation is persistent, and below we look for potential limits to this arbitrage.

The spike in the GBP basis coincides with important events leading up to Brexit. Brexit is a term for the process dealing with United Kingdom's exit from the European Union. The process started in February 2016 when British Prime Minister David Cameron announced that a referendum regarding membership in the European Union was taking place later the same year. At this point, we see a large spike in the forward basis. In June the same year, the referendum was held and the British people voted to exit the European Union, causing another spike in the forward basis. The positive basis remains persistent and does not cross zero before 2018. In 2019, the basis moved upwards again, potentially due to uncertainties surrounding the final Brexit deal between United Kingdom and the European Union which was originally scheduled to take place in March or April 2019, but was postponed to October 2019.

A more interesting aspect from a financial perspective is what economic factors drove the basis upwards. The persistence of the arbitrage cannot be explained by traditional credit risk measures or open interest of the contracts. LIBOR-OIS spread did not markedly increase in the period of 2016-2019. In fact, this spread was higher during the financial crisis, a time in which the forward basis traded at levels lower than Brexit levels. The Bank for International Settlements provides data on open interest in both over-the-counter and exchange-traded markets across the world. Data[12] shows that volume in FRAs linked to GBP decreased significantly from 2014 with only small reductions from 2016 and onwards.

Liquidity in GBP FRAs worsened after important Brexit events. Using the relative bid-ask spread of the 6x12 FRA, we plot the relative bid-ask spread and the forward basis below. The relationship between the forward basis and an increased level of the relative bid-ask spread is apparent. This suggests that high trading costs related to the arbitrage strategy could be an important limit for arbitrageurs seeking to exploit the opportunity.

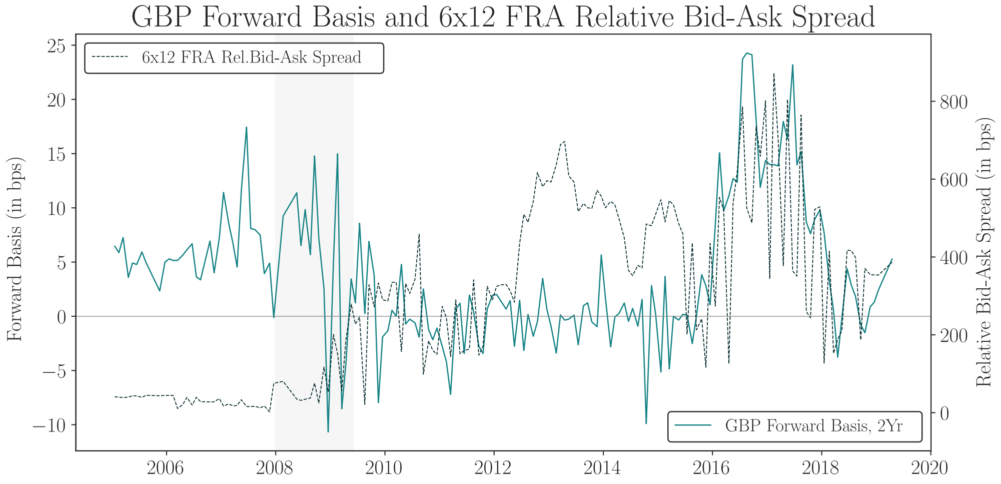


Figure 7: GBP Forward Basis and Relative Bid-Ask Spread

[12] See for example the dataset "OTC derivatives outstanding" on https://www.bis.org/statistics/full_data_sets.htm

7.2 The Long-Tenor Futures Basis Arbitrage

In contrast to the GBP forward basis arbitrage, the futures bases for longer tenors have traded at large positive levels during most of our historical sample. This futures basis arbitrage have been treated in the literature before, most notably by Gupta and Subrahmanyam (2000). It has persisted for nearly 30 years, and is worth some attention. This arbitrage is more of a near-arbitrage compared to the FRA replication portfolio. Despite this, we should not expect to see such high levels without any consistent factor explaining away most of the basis.

The means for the 5-, 7-, and 10-year USD futures basis in the post-crisis have been 5.6 bps, 10.7 bps and 17.4 bps respectively. Similar values were found in the other periods. At first sight this looks like arbitrage opportunities. We saw in the replication portfolio that traders were able to closely replicate the cash flows of the floating leg in an IRS. This begs the question why this arbitrage opportunity exist.

Gupta and Subrahmanyam (2000) put forth nearly 20 years ago that convexity is the main limit to arbitrage for the futures basis. Other factors like credit risk or liquidity were found to be either statistically or economically insignificant.

We use a similar procedure, albeit with other proxies for credit risk and liquidity. Our results are remarkably similar, showing the survival of their findings in the post-crisis period. The convexity bias explains away most arbitrage profits, showing the robustness of Gupta and Subrahmanyam's initial finding. This is surprising due to several factors. The mark-to-market feature of futures contracts that leads to the convexity adjustments and the collateralization procedures of most interest rate swaps are more similar today relative to 20 years ago. Following from this, convexity should matter less, not more or equally much as before. That could be one of the explanations for why we are finding that the convexity bias is not able to explain the variation in the futures basis well, only the mean level. This is in contrast to what Gupta and Subrahmanyam find.

To test the effect of credit risk on the futures basis arbitrage, we run regressions with the basis for tenors 5, 7, and 10 as dependent variables and the LIBOR-OIS spread, defined as three-month IBOR minus three-month OIS, as the independent variable. To capture the effect of credit risk, and not the effect of discounting, we use futures bases with OIS discounting. We run regressions on levels and first differences and table 7 in the appendix reports the results. The LIBOR-OIS spread is statistically significant at the 5%-level, so a linear relationship exists, but it is not strong. Its economical significance is small. For a 1 bps change in LIBOR-OIS spread, the futures basis for the 5-year swap changes by 0.15 bps[13]. Similar results are found in both crisis and post-crisis periods, suggesting that credit risk is economically insignificant even in periods of market stress seemingly in line with both Bomfim (2003).

For liquidity, we run the bases against tenor basis swaps with legs of one- and three-month IBORs. Using first differences, we do not find any statistically significant linear relationship between liquidity and the basis at any level, neither during the crisis nor post-crisis. Results for the 5-year basis are reported in table 8 in the appendix. Similar results were found for 7- and 10-year tenors.

Table 4 below reports the summary statistics of the USD futures bases before and after convexity bias is taken into account. Most notable, all means converge towards no-arbitrage levels after we account for convexity. However, the standard deviations stay at similar levels, suggesting that the convexity is able to reduce the mean, but not explain the variation well. After accounting for convexity on the 10-year basis, more than 25% of the observations have an absolute value of more than 6 bps, an economically significant difference for IRSs with notionals above $100 Mn.

However, relative to other near-arbitrage strategies, this futures basis arbitrage for longer tenors require an enormous amount of maintenance, both in form of margin requirements to be met, and waiting period for the payoff. The notionals that need to be locked up to obtain large economic profits are potentially large, a tighter regulations in the post-crisis environment may play a key role in the ability of arbitrageurs to profit from small arbitrage opportunities (Boyarchenko et al., 2018).

[13] As a comparison, similar regressions with the convexity bias as independent variable suggest changes of 3 bps to 6 bps.

	2Yr, no Convexity	3Yr, no Convexity	5Yr, no Convexity	7Yr, no Convexity	10Yr, no Convexity
Count	2391.0000	2391.0000	2391.0000	2391.0000	2391.0000
Mean	0.6193	1.6735	5.5653	10.7449	17.3677
Std	2.6184	3.3431	4.4783	5.6701	7.9281
Min	-12.1263	-16.5098	-14.0029	-7.8528	-0.5634
25%	-0.6244	-0.1810	2.6958	6.9447	11.3156
50%	0.4874	1.4484	5.2324	10.0155	15.7785
75%	1.7153	3.2524	7.9891	14.2035	23.1105
Max	18.4315	21.9340	28.4668	35.8580	45.6846

(a) USD Futures Basis without Convexity Adjustments, Post-Crisis Period

	2Yr, w/Convexity	3Yr, w/Convexity	5Yr, w/Convexity	7Yr, w/Convexity	10Yr, w/Convexity
Count	2387.0000	2387.0000	2387.0000	2387.0000	2387.0000
Mean	-0.1774	-0.0644	1.0694	2.5635	2.4810
Std	2.6361	3.3957	4.4598	5.2050	6.2073
Min	-12.5047	-19.0901	-20.6756	-19.9815	-18.8926
25%	-1.4355	-1.8941	-1.6091	-0.7102	-1.5009
50%	-0.3015	-0.2682	0.9883	2.3010	1.6051
75%	0.9810	1.6049	3.5145	5.5457	6.3103
Max	17.9088	20.7995	25.5346	27.0236	28.2532

(b) USD Futures Basis with Convexity Adjustments, Post-Crisis Period

Table 4: USD Futures Basis with and without Convexity

8 Conclusion

Our goal in this thesis was first to provide an updated framework of computing futures- and forward basis. We provided a flexible method that allowed us to compute bases for three markets. As a contribution to literature, we have shown how forward rate agreements, OIS discounting and tenor basis swaps can be included in the computations. In addition, we have tried to make the process as open as possible by putting forth assumptions taken in addition to including the code for computation of the bases in the appendix. All of this is fully implemented in Python, and requires no costly programming languages with exception of access to the relevant data.

Secondly, we updated and extended previous empirical research to include more recent basis-levels. It was important for us to include the forward bases as it has to a lesser extent been empirically examined. Our focus was centered around presenting an overview of the patterns of the bases in addition to including some correlation measures both across instruments and across markets. We showed that despite markets being mostly efficient, there exist deviations from the theory of the replication portfolios constructed. Furthermore, we explored two specific arbitrage strategies and discussed drivers of limits to arbitrage in these specific strategies.

Limitations of the thesis include a shallow treatment of some of the practical considerations for arbitrageurs wanting to trade on these strategies. It is not unreasonable to assume that the arbitrage opportunities presented could be eaten away by trading costs and restricted access to capital.

Further research could look more into correlations, in particular the collapse of correlations between USD and EUR bases in post crisis, or the GBP forward and futures basis in the post-crisis period. Alternatively, the behavior of the bases was not well captured by any of our factors, and this also an interesting avenue for further research.

References

Aikin, S. (2012). *STIR futures: Trading Euribor and Eurodollar Futures.* Harriman House Limited.

Ametrano, F., & Bianchetti, M. (2013). Everything You Always Wanted to Know about Multiple Interest Rate Curve Bootstrapping but Were Afraid to Ask. *SSRN Electronic Journal.* Retrieved from `https://doi.org/10.2139/ssrn.2219548` doi: 10.2139/ssrn.2219548

Ballabio, L., Ametrano, F. M., Spanderen, K., Caspers, P., , Pepe, Mazzocchi, P., ... Eddelbuettel, D. (2019). *lballabio/QuantLib: 1.15.* Zenodo. doi: 10.5281/zenodo.2572597

Ballabio, L., & Balaraman, G. (2017). *QuantLib Python Cookbook.* Leanpub.

Bank for International Settlements. (2019). Statistical Release: OTC Derivatives Statistics at End-December 2018. Retrieved from `https://www.bis.org/publ/otc_hy1905.pdf`

Bloomberg Quantitative Analytics. (2017). OIS Discounting and Dual-Curve Stripping Methodology.

Bomfim, A. N. (2003). Counterparty Credit Risk in Interest Rate Swaps During Times of Market Stress.

Boyarchenko, N., Eisenbach, T. M., Gupta, P., Shachar, O., & Van Tassel, P. (2018). Bank-Intermediated Arbitrage. *FRB of New York Staff Report No. 858*.

Burghardt, G. (2003). *The Eurodollar Futures and Options Handbook.* McGraw Hill Professional.

Collin-Dufresne, P., & Solnik, B. (2001). On the Term Structure of Default Premia in the Swap and LIBOR Markets. *The Journal of Finance*, *56*(3), 1095–1115.

Cox, J. C., Ingersoll Jr, J. E., & Ross, S. A. (1981). The Relation Between Forward Prices and Futures Prices. *Journal of Financial Economics*, *9*(4), 321–346.

Cox, J. C., Ingersoll Jr, J. E., & Ross, S. A. (2005). A Theory of the Term Structure of Interest Rates. In *Theory of valuation* (pp. 129–164). World Scientific.

Darbyshire, J. (2016). *The Pricing and Trading of Interest Rate Derivatives: A Practical Guide to Swaps.* Aitch & Dee Limited.

Du, W., Tepper, A., & Verdelhan, A. (2018). Deviations from Covered Interest Rate Parity. *The Journal of Finance*, *73*(3), 915–957.

Duarte, J., Longstaff, F. A., & Yu, F. (2006). Risk and Return in Fixed-Income Arbitrage: Nickels in Front of a Steamroller? *The Review of Financial Studies*, *20*(3), 769–811.

Feldhütter, P., & Lando, D. (2008). Decomposing Swap Spreads. *Journal of Financial Economics*, *88*(2), 375–405.

Grinblatt, M., & Jegadeesh, N. (1996). Relative Pricing of Eurodollar Futures and Forward Contracts. *The Journal of Finance*, *51*(4), 1499–1522.

Gupta, A., & Subrahmanyam, M. G. (2000). An Empirical Examination of the Convexity Bias in the Pricing of Interest Rate Swaps. *Journal of Financial Economics*, *55*(2), 239–279.

Gurrieri, S., Nakabayashi, M., & Wong, T. (2009). Calibration Methods of Hull-White Model. Retrieved from `https://pdfs.semanticscholar.org/b300/5e0e59a60bf7bb9b1e4646f96124bc7b8e16.pdf`

Henrard, M. (2009). Efficient Swaptions Price in Hull-White One Factor Model. *arXiv preprint arXiv:0901.1776*.

Hou, D., & Skeie, D. R. (2014). LIBOR: Origins, Economics, Crisis, Scandal, and Reform. *Federal Reserve Bank of New York Staff Reports*.

Hull, J. (2017). Technical Note No. 1, Options, Futures, and Other Derivatives, Convexity Adjustments to Eurodollar Futures. Retrieved from `http://www-2.rotman.utoronto.ca/~hull/TechnicalNotes/TechnicalNote1.pdf`

Hull, J. (2018). *Options, Futures, and other Derivatives*. Pearson Education.

Hull, J., & White, A. (1990). Pricing Interest-Rate-Derivative Securities. *The Review of Financial Studies*, *3*(4), 573–592.

Hull, J., & White, A. (1994). Numerical Procedures for Implementing Term Structure Models I: Single-Factor Models. *Journal of Derivatives*, *2*(1), 7–16.

Hull, J., & White, A. (2001). The General Hull–White Model and Supercalibration. *Financial Analysts Journal*, *57*(6), 34–43.

Hull, J., & White, A. (2013). LIBOR vs. OIS: The Derivatives Discounting Dilemma. *Journal Of Investment Management*, *11*(3), 14–27.

Jarrow, R. A., & Oldfield, G. S. (1981). Forward Contracts and Futures Contracts. *Journal of Financial Economics*, *9*(4), 373–382.

Johannes, M., & Sundaresan, S. (2007). The Impact of Collateralization on Swap Rates. *The Journal of Finance*, *62*(1), 383–410.

Katajamäki, M. (2017). QuantLib : Hull-White one-factor model calibration . Retrieved from `http://mikejuniperhill.blogspot.com/2017/09/quantlib-hull-white-one-factor-model.html`

Labuszewski, J. W., Nyhoff, J. E., Peterson, P. E., & CME Group. (2010).

The CME Group Risk Management Handbook: Products and Applications (Vol. 410). John Wiley & Sons.

Liu, J., Longstaff, F. A., & Mandell, R. E. (2006). The Market Price of Risk in Interest Rate Swaps: The Roles of Default and Liquidity Risks. *The Journal of Business*, *79*(5), 2337–2359.

Mercurio, F. (2009). Interest Rates and the Credit Crunch: New Formulas and Market Models. *Bloomberg Portfolio Research Paper*(2010-01).

Minton, B. A. (1997). An Empirical Examination of Basic Valuation Models for Plain Vanilla US Interest Rate Swaps. *Journal of Financial Economics*, *44*(2), 251–277.

OpenGamma. (2013). Interest Rate Instruments and Market Conventions Guide. Retrieved from `https://developers.opengamma.com/quantitative-research/Interest-Rate-Instruments-and-Market-Conventions.pdf`

Ron, U. (2000). *A Practical Guide to Swap Curve Construction.* Citeseer.

Shleifer, A., & Vishny, R. W. (1997). The Limits of Arbitrage. *The Journal of finance*, *52*(1), 35–55.

Smith, D. J. (2013, May). Valuing Interest Rate Swaps Using Overnight Indexed Swap (OIS) Discounting. *The Journal of Derivatives*, *20*(4), 49–59. doi: 10.3905/jod.2013.20.4.049

Sokol, A. (2014). *Long-Term Portfolio Simulation: For XVA, Limits, Liquidity and Regulatory Capital.* Risk Books.

Vasicek, O. (1977). An Equilibrium Characterization of the Term Structure. *Journal of financial economics*, *5*(2), 177–188.

Veronesi, P. (Ed.). (2016). *Handbook of Fixed-Income Securities.* John Wiley & Sons, Inc. doi: 10.1002/9781118709207

White, R. (2012). Multiple Curve Construction. *OpenGamma Quantitative Research.*

A Appendix

A.1 Supporting Tables

	Pre-Crisis USD	Pre-Crisis EUR	Pre-Crisis GBP
Count	720.0000	1788.0000	1767.0000
Mean	-0.9424	-0.3183	5.8308
Std	2.4139	1.9147	3.8113
Min	-16.4285	-25.3972	-22.7776
25%	-1.3330	-0.6390	4.4071
50%	-0.8355	-0.1684	5.6957
75%	-0.2944	0.1895	7.2832
Max	14.9328	12.9445	24.3809

(a) Pre-Crisis Forward Basis, Daily Data

	Crisis USD	Crisis EUR	Crisis GBP
Count	760.0000	767.0000	760.0000
Mean	-0.6883	-0.5635	4.2704
Std	6.9474	4.1942	6.0336
Min	-33.0276	-25.9814	-19.5868
25%	-4.6703	-2.8223	0.1816
50%	-0.8952	-0.5680	5.0099
75%	2.8504	1.4486	8.4852
Max	33.6223	21.2174	25.2247

(b) Crisis Forward Basis, Daily Data

	Post-Crisis USD	Post-Crisis EUR	Post-Crisis GBP
Count	2391.0000	2423.0000	2382.0000
Mean	-0.0430	-0.1023	3.7169
Std	2.8025	1.8772	6.4506
Min	-15.3381	-21.2994	-12.4831
25%	-1.4115	-0.6426	-0.2736
50%	-0.0823	-0.0826	1.2916
75%	1.2289	0.4796	5.4370
Max	19.0340	17.3626	25.7842

(c) Post-Crisis Forward Basis, Daily Data

Table 5: Summary Statistics, 2-Year Forward Basis

	Pre-Crisis USD	Pre-Crisis EUR	Pre-Crisis GBP
Count	720.0000	1788.0000	1767.0000
Mean	0.2624	0.2015	-0.0840
Std	2.1813	1.5972	1.5899
Min	-16.1409	-12.2832	-10.6190
25%	-0.4516	-0.5871	-0.8258
50%	0.3245	0.2021	-0.0605
75%	1.1222	1.0056	0.7141
Max	16.0300	10.5921	13.3493

(a) Pre-Crisis Futures Basis, Daily Data

	Crisis USD	Crisis EUR	Crisis GBP
Count	760.0000	767.0000	760.0000
Mean	0.6595	0.3191	-0.0761
Std	6.4580	4.2317	4.7036
Min	-29.1625	-24.7040	-22.9400
25%	-3.1490	-1.8503	-2.6309
50%	0.3316	0.1930	-0.1451
75%	3.7583	1.9723	2.5234
Max	37.3810	24.4225	20.7323

(b) Crisis Futures Basis, Daily Data

	Post-Crisis USD	Post-Crisis EUR	Post-Crisis GBP
Count	2391.0000	2423.0000	2382.0000
Mean	0.6407	0.2425	0.4805
Std	2.6214	1.8889	1.9404
Min	-11.9754	-17.3593	-10.4513
25%	-0.6031	-0.4067	-0.5218
50%	0.5022	0.1089	0.4229
75%	1.7413	0.7153	1.4413
Max	18.6262	17.4476	13.1402

(c) Post-Crisis Futures Basis, Daily Data

Table 6: Summary Statistics, 2-Year Futures Basis

	Time-Period	
	Crisis	Post-Crisis
	(1)	(2)
LIBOR-OIS Spread	0.018***	-0.03***
	(0.005)	(0.009)
Constant	5.009***	6.229***
	(0.417)	(0.206)
Observations	760.0	2384.0
R2	0.019	0.005
Notes:		***p<0.01, **p<0.05, *p<0.10
		Futures basis and LIBOR-OIS spread in bps

(a) Levels
Dependent Variable: 5-Year USD Futures Basis with OIS Discounting

	Time-Period	
	Crisis	Post-Crisis
	(1)	(2)
ΔLIBOR-OIS Spread	0.143**	0.15**
	(0.069)	(0.071)
Constant	0.018	-0.002
	(0.365)	(0.109)
Observations	759.0	2383.0
R2	0.006	0.002
Notes:		***p<0.01, **p<0.05, *p<0.10
		Futures basis and LIBOR-OIS spread in bps

(b) First Differences
Dependent Variable: Δ5-Year USD Futures Basis with OIS Discounting

Table 7: Regression Table - Credit Risk

	Time-Period	
	Crisis	Post-Crisis
	(1)	(2)
Tenor Basis 5Yr, 1M vs. 3M	0.469*** (0.072)	0.081*** (0.029)
Constant	3.841*** (0.458)	4.767*** (0.3)
Observations	760.0	2391.0
R2	0.052	0.003
Notes:		***p<0.01, **p<0.05, *p<0.10 Futures basis and tenor basis in bps

(a) Levels
Dependent Variable: 5-Year USD Futures Basis with OIS Discounting

	Time-Period	
	Crisis	Post-Crisis
	(1)	(2)
ΔTenor Basis 5Yr, 1M vs. 3M	-0.105 (0.691)	0.103 (0.279)
Constant	0.019 (0.366)	-0.0 (0.109)
Observations	759.0	2390.0
R2	0.0	0.0
Notes:		***p<0.01, **p<0.05, *p<0.10 Futures basis and tenor basis in bps

(b) First Differences
Dependent Variable: Δ5-Year USD Futures Basis with OIS Discounting

Table 8: Regression Table - Liquidity and Optionality

A.2 Supporting Figures

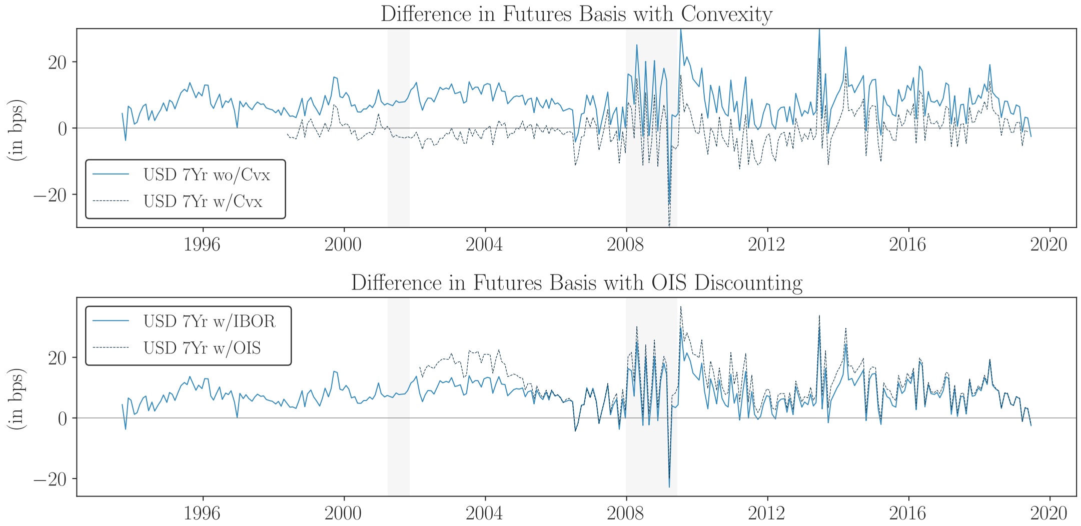
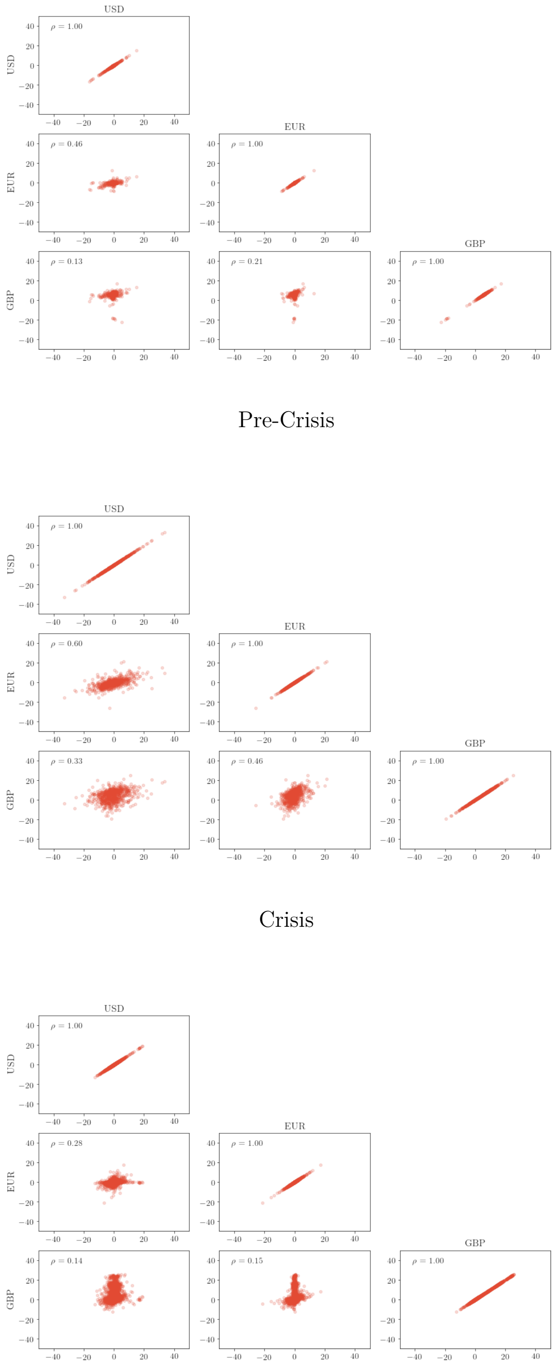


Figure 8: Difference in Futures Basis with OIS Discounting and Convexity Adjustments

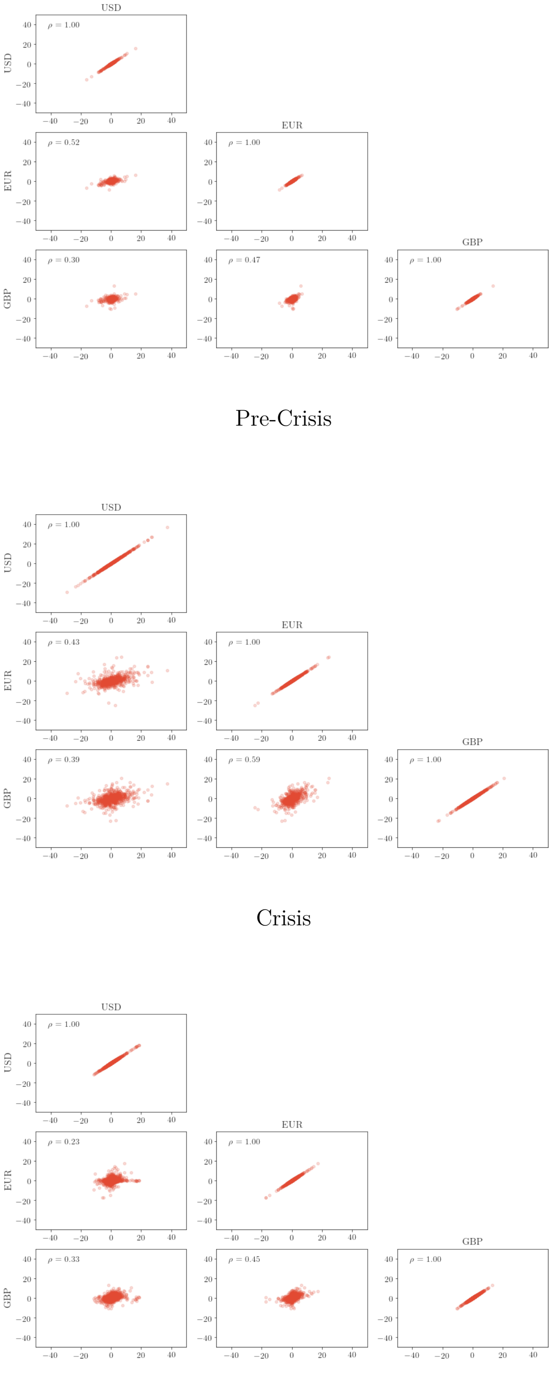


Pre-Crisis

Crisis

Post-Crisis

Figure 9: Scatter Plots of Forward Basis for Different Currencies



Pre-Crisis

Crisis

Post-Crisis

Figure 10: Scatter Plots of Futures Basis for Different Currencies

A.3 Data Sources

Table 9: Conventions for Interest Rate Swaps in Currencies USD, EUR and GBP

	USD	**EUR**	**GBP**
Bloomberg Code	USSWAPx Curncy	EUSAx Curncy	BPSWx Curncy
Index	3M USD LIBOR	6M EURIBOR	6M GBP LIBOR
Settlement	T+2 Days	T+2 Days	T+0 Days
Calendar	United States	Target	England
Bus Adj	Modified Following	Modified Following	Modified Following
Roll Conv	Backward (EOM)	Backward (EOM)	Backward (EOM)
Float Leg: Day Count	ACT/360	ACT/360	ACT/365
Float Leg: Pay Freq	Quarterly	Semiannual	Semiannual
Fixed Leg: Day Count	30/360	30/360 (Bond Basis)	ACT/365
Fixed Leg: Pay Freq	Semiannual	Annual	Semiannual
Discounting	OIS	OIS	OIS

Table 10: Bloomberg Codes for Data used in Computations

	USD	**EUR**	**GBP**
Deposit Rates	US0003M Index	EUR003M Index EUR006M Index	BP003M Index BP006M Index
STIR-Futures	ED1 Comdty ... ED40 Comdty	ER1 Comdty ... ER24 Comdty	L 1 Comdty ... L 24 Comdty
FRAs	USFR0CF Curncy USFR0FI Curncy USFR0I1 Curncy USFR011C Curncy USFR1C1F Curncy USFR1F1I Curncy USFR1I2 Curncy	EUR006M Index EUFR0F1 Curncy EUFR011F Curncy EUFR1F2 Curncy	BP0006M Index BPFR0F1 Curncy BPFR011F Curncy BPFR1F2 Curncy
OIS Rates	USSOC Curncy USSOF Curncy USSOI Curncy USSO1 Curncy USSO1F Curncy USSO2 Curncy USSO3 Curncy USSO4 Curncy USSO5 Curncy	EUSWEC Curncy EUSWEF Curncy EUSWEI Curncy EUSWE1 Curncy EUSWE1F Curncy EUSWE2 Curncy EUSWE3 Curncy EUSWE4 Curncy EUSWE5 Curncy EUSWE7 Curncy	BPSWSC Curncy BPSWSF Curncy BPSWSI Curncy BPSWS1 Curncy BPSWS1F Curncy BPSWS2 Curncy BPSWS3 Curncy BPSWS4 Curncy BPSWS5 Curncy BPSWS7 Curncy
IRSs	USSWAP2 Curncy USSWAP3 Curncy USSWAP5 Curncy USSWAP7 Curncy USSWAP10 Curncy	EUSA2 Curncy EUSA3 Curncy EUSA5 Curncy EUSA7 Curncy	BPSW2 Curncy BPSW3 Curncy BPSW5 Curncy BPSW7 Curncy
Swaptions	USSN019 Curncy USSN028 Curncy USSN037 Curncy USSN046 Curncy USSN055 Curncy USSN064 Curncy USSN073 Curncy USSN082 Curncy USSN091 Curncy USSV019 Curncy USSV028 Curncy USSV037 Curncy USSV046 Curncy USSV055 Curncy USSV064 Curncy USSV073 Curncy USSV082 Curncy USSV091 Curncy	EUSN019 Curncy EUSN028 Curncy EUSN037 Curncy EUSN046 Curncy EUSN055 Curncy EUSN064 Curncy EUSN073 Curncy EUSN082 Curncy EUSN091 Curncy	
Other	USBG7 Curncy USBG10 Curncy USBAAC Curncy	EUBSV2 Curncy EUBSV3 Curncy EUBSV5 Curncy EUBSV7 Curncy	BPSFVC2 Curncy BPSFVC3 Curncy BPSFVC5 Curncy BPSFVC7 Curncy

A.4 Calibration of Vasicek Model

The Vasicek model can be written as

$$dr = \kappa(\mu - r)dt + \sigma dz$$

Using a time series of three-month (or six-month) deposit rates denoted by r_t, we can calibrate parameters κ, μ and σ by the following regression model

$$r_{t+1} = a + br_t + \epsilon_t$$

where $a = \mu(1 - e^{-\kappa\Delta t})$ and $b = e^{-\kappa\Delta t}$.

After running this OLS regression, parameters can be backed out by using

$$\kappa = -\frac{ln(b)}{\Delta t}$$

$$\mu = \frac{a}{1 - e^{-k\Delta t}}$$

$$\sigma = std(\epsilon)\sqrt{\frac{2\kappa}{1 - e^{-2\kappa\Delta t}}}$$

The next steps involves finding theoretical futures and forward rates under the Vasicek process with the calibrated parameters. Grinblatt and Jegadeesh (1996) provide a clear overview of this process. The equations below are based on their paper.

Vasicek (1977) shows that prices of pure discount bonds are given by

$$P(s,t) = a(s-t)exp(-b(s-t)r(s)$$

where s and t indicate time, and is measured in year-fractions (for example ACT/360) and

$$b(x) = (1/\kappa)[1 - exp(-\kappa x)]$$

$$a(x) = exp\left[(b(x) - x)(\mu - \sigma^2/2\kappa^2) - (\sigma^2/4\kappa)b(x)^2\right]$$

The prices of pure discount bonds imply that the Vasicek forward rate is given by

$$f(T1,T2) = d(T1,T2)\left[\frac{P(0,T1)}{P(0,T2)} - 1\right] = d(T1,T2)\left[\frac{a(T1)}{a(T2)}exp[(b(T2) - b(T1))r(0)]\right]$$

where $d(s,t)$ is 360 divided by number of days between s and t.

The futures rate requires a few more steps, and is given by

$$F(T1, T2) = d(T1, T2)\left[(1/a(T2 - T1))E[exp(b(T2 - T1)r(T1))] - 1\right]$$

where

$$E[exp(b(T2-T1)r(T1))] = exp[B(T2-T1)E[r(T1)] + \frac{1}{2}B(T2-T1)^2 var[r(T1)]]$$

$$E[r(T1)] = exp(\kappa T1)r(T2) + [1 - exp(-\kappa T1)]\mu^*$$

$$var[r(T1)] = \frac{\sigma^2[1 - exp(-2\kappa T1]}{2\kappa}$$

Once both theoretical futures rates and forward rates are derived, the convexity adjustment under the Vasicek model is given by

$$cvx^{Vasicek} = F(T1, T2) - f(T1, T2)$$

A.5 Python Code for Computing the Futures Basis

```

# coding: utf 8

# In[ ]:


# IMPORT LIBRARIES
import math
import numpy as np
import pandas as pd
import seaborn as sns
import QuantLib as ql
from scipy import stats
from functools import reduce
from datetime import datetime
from itertools import compress
from collections import namedtuple
from scipy.interpolate import interp1d


# In[ ]:


# CHOOSE CURRENCY AND WHETHER YOU WANT CONVEXITY ADJUSTMENT
```

```
currency = 'USD'
adjustCvx = True

#...based on these inputs, we set some
if currency == 'USD':
    #...define number of days after trade that deal settles
    settleDays = 2
    #...specify calendar for market
    cal = ql.UnitedStates()
    #...specify day count convention
    dayCount = ql.Actual360()
    dayCountFlt = ql.Actual360()
    dayCountFix = ql.Thirty360()
    #...define frequency of floating payments and other date rules
    freq = ql.Quarterly
    freqFlt = 4
    conv = ql.M
    dtRule = ql.
    EOM = True
    #...define frequency of fixed payments
    freqFixed = 2
    #...filename for input file
    filename = 'usd_data_June.xlsx'
    #...filename for output file
    excelFilename = 'USD Implied IRS '+str(ql.Date.todaysDate())+'.xlsx'
    #...filename for input file with convexity information
    filenameCvx = 'swaption_usd.xlsx'

if currency == 'EUR':
    #...define number of days after trade that deal settles
    settleDays = 2
    #...specify calendar for market
    cal =  ql.TARGET()
    #...specify day count convention
    dayCount = ql.Actual360()
    dayCountFlt = ql.Actual360()
    dayCountFix = ql.Thirty360(ql.Thirty360.BondBasis)
    #...define frequency of floating payments and other date rules
    freq = ql.Quarterly
    freqFlt = 4
    conv = ql.ModifiedFollowing
    dtRule = ql.DateGeneration.Backward
    EOM = True
    #...define frequency of fixed payments
    freqFixed = 1
    #...filename for input file
    filename = 'eur_data_June.xlsx'
    #...filename for output file
    excelFilename = 'EUR Implied IRS '+str(ql.Date.todaysDate())+'.xlsx'
    #...filename for input file with convexity information
    filenameCvx = 'swaption_eur.xlsx'

if currency == 'GBP':
    #...define number of days after trade that deal settles
    settleDays = 0
    #...specify calendar for market
    cal = ql.UnitedKingdom()
    #...specify day count convention
    dayCount = ql.Actual365Fixed()
    dayCountFlt = ql.Actual365Fixed()
    dayCountFix = ql.Actual365Fixed()
    #...define frequency of floating payments and other date rules
    freq = ql.Quarterly
    freqFlt = 4
    conv = ql.ModifiedFollowing
    dtRule = ql.DateGeneration.Backward
    EOM = True
    #...define frequency of fixed payments
    freqFixed = 2#4
    #...filename for input file
    filename = 'gbp_data.xlsx'
    #...filename for output file
    excelFilename = 'GBP Implied IRS '+str(ql.Date.todaysDate())+'.xlsx'


# In[ ]:


# IMPORT DATA
libor_rates = pd.read_excel(filename,sheet_name='ibor',index_col='Dates')
fut_prices = pd.read_excel(filename,sheet_name='futures',index_col='Dates')
ois_rates = pd.read_excel(filename,sheet_name='ois',index_col='Dates')
swap_rates = pd.read_excel(filename,sheet_name='swaps',index_col='Dates')
```

```
fwd_rates = pd.read_excel(filename,sheet_name='forwards',index_col='Dates')
if ((currency=='USD') | (currency=='EUR')) & (adjustCvx==True):
    swap_rates_HW = pd.read_excel(filenameCvx,sheet_name='swaps',index_col='Dates')
    swaption_norm = pd.read_excel(filenameCvx,sheet_name='normal',index_col='Dates')
    if currency == 'USD':
        swaption_black = pd.read_excel(filenameCvx,sheet_name='lognormal',index_col='Dates')


# ### Compute Date Schedules for Futures and Swaps

# In[ ]:


# DEFINE USER DEFINED FUNCTIONS
def createPmtSchedule(settle,tenor,maxTenor):
    pmtList = []
    if tenor <= maxTenor:
        maturityDate = cal.advance(settle, np.int(tenor), ql.Years)
        #...compute quarterly payment schedule between settle and maturity
        swpPmtSchedule = ql.Schedule(settle, maturityDate, ql.Period(freq),
                                    cal, conv, conv, dtRule, EOM)
        #...extract payment dates as a list
        pmtDates = list(swpPmtSchedule)
        #...adjust pmtDates so that we get actual payment dates
        pmtDates = pmtDates[ tenor*freqFlt:]
        #...append results to list and convert to non QuantLib format
        pmtList.append([datetime(dt.year(), dt.month(), dt.dayOfMonth()) for dt in pmtDates])
    else:
        pmtList.append(np.ones(tenor*freqFlt)*np.nan)
    return pmtList;

def yearFractionSwapPayments(settle,tenor,maxTenor,daycount):
    yfList = []
    for dt in createPmtSchedule(settle,tenor,maxTenor)[0]:
        try:
            yfList.append(daycount.yearFraction(settle,ql.Date(dt.day,dt.month,dt.year)))
        except:
            yfList.append(np.nan)
    return yfList;


# In[ ]:


# Create lists to store dates
depoValue = []
futuresValue = []
futuresEnd = []
swpPmtDates2 = []
swpPmtDates3 = []
swpPmtDates5 = []
swpPmtDates7 = []
swpPmtDates10 = []
yfDeposit = []
yfFuturesVal = []
yfFuturesEnd = []
yfSwapFlt2 = []
yfSwapFlt3 = []
yfSwapFlt5 = []
yfSwapFlt7 = []
yfSwapFlt10 = []
yfSwapFix2 = []
yfSwapFix3 = []
yfSwapFix5 = []
yfSwapFix7 = []
yfSwapFix10 = []

futuresExpiry = []

for dt, row in fut_prices.iterrows():

    #############################
    # SET UP GENERAL PARAMETERS #
    #############################
    #...convert date to QuantLib format
    tradeDate = ql.Date(dt.day,dt.month,dt.year)
    #...check how many years we can compute swap rates for in the specific row
    maxTenor = np.int(np.floor((~row.isna()).sum()/4))
    #...set settle date at T+settleDays days
    settleDate = cal.advance(tradeDate, np.int(settleDays), ql.Days, conv)
```

```

 ##############################################
 # DATE SCHEDULES FOR DEPOSITS AND FUTURES #
 ##############################################
 #...compute value dates for 3 month deposit
 depoDates = cal.advance(settleDate, 3, ql.Months, conv)
 #...compute value dates for futures contracts
 futDates = [ql.IMM.nextDate(cal.advance(settleDate,  1, ql.Days,
conv))]#[ql.IMM.nextDate(tradeDate)]
 for mnth in range(maxTenor*4 1):
     futDates.append(ql.IMM.nextDate(futDates[ 1]))
 #...compute end dates for futures contracts (as defined in figure
  7 from Ametrano & Bianchetti)
 futEndDates = [cal.advance(valDate, 3, ql.Months, conv) for
valDate in futDates]
 #...add results to lists
 depoValue.append([datetime(depoDates.year(), depoDates.month(),
depoDates.dayOfMonth())])
 futuresValue.append([datetime(dt.year(), dt.month(), dt.
dayOfMonth()) for dt in futDates])
 futuresEnd.append([datetime(dt.year(), dt.month(), dt.dayOfMonth
()) for dt in futEndDates])

 futExpiryDates = [cal.advance(dt,  2, ql.Days, conv) for dt in
futDates]
 futuresExpiry.append([datetime(dt.year(), dt.month(), dt.
dayOfMonth()) for dt in futExpiryDates])

 ################################################
 # PAYMENT SCHEDULES FOR INTEREST RATE SWAPS #
 ################################################
 swpPmtDates2.append(createPmtSchedule(settleDate,2,maxTenor)[0])
 swpPmtDates3.append(createPmtSchedule(settleDate,3,maxTenor)[0])
 swpPmtDates5.append(createPmtSchedule(settleDate,5,maxTenor)[0])
 swpPmtDates7.append(createPmtSchedule(settleDate,7,maxTenor)[0])
 swpPmtDates10.append(createPmtSchedule(settleDate,10,maxTenor)
[0])

 ##############################################
 # YEAR FRACTIONS FOR DEPOSITS AND FUTURES #
 ##############################################
 #...compute the year fraction between settle date and 3 month
deposit rate (using dayCount variable)
 yfDep = dayCount.yearFraction(settleDate,depoDates)
 #...compute the year fraction between settle date and futures
```

```
value date (using dayCount variable)
 yfFutVal = [dayCount.yearFraction(settleDate,dt) for dt in futDates]
 #...compute the year fraction between settle date and futures end date (using dayCount variable)
 yfFutEnd = [dayCount.yearFraction(settleDate,dt) for dt in futEndDates]
 #...append year fractions to lists
 yfDeposit.append(yfDep)
 yfFuturesVal.append(yfFutVal)
 yfFuturesEnd.append(yfFutEnd)

 ##########################################
 # YEAR FRACTIONS FOR INTEREST RATE SWAPS #
 ##########################################
 #...2 year tenor
 yfSwpFlt2 = yearFractionSwapPayments(settleDate,2,maxTenor,dayCountFlt)
 yfSwpFix2 = yearFractionSwapPayments(settleDate,2,maxTenor,dayCountFix)
 #...3 year tenor
 yfSwpFlt3 = yearFractionSwapPayments(settleDate,3,maxTenor,dayCountFlt)
 yfSwpFix3 = yearFractionSwapPayments(settleDate,3,maxTenor,dayCountFix)
 #...5 year tenor
 yfSwpFlt5 = yearFractionSwapPayments(settleDate,5,maxTenor,dayCountFlt)
 yfSwpFix5 = yearFractionSwapPayments(settleDate,5,maxTenor,dayCountFix)
 #...7 year tenor
 yfSwpFlt7 = yearFractionSwapPayments(settleDate,7,maxTenor,dayCountFlt)
 yfSwpFix7 = yearFractionSwapPayments(settleDate,7,maxTenor,dayCountFix)
 #...10 year tenor
 yfSwpFlt10 = yearFractionSwapPayments(settleDate,10,maxTenor,dayCountFlt)
 yfSwpFix10 = yearFractionSwapPayments(settleDate,10,maxTenor,dayCountFix)
 #...append year fractions to lists
 yfSwapFlt2.append(yfSwpFlt2)
 yfSwapFlt3.append(yfSwpFlt3)
 yfSwapFlt5.append(yfSwpFlt5)
 yfSwapFlt7.append(yfSwpFlt7)
```

```
    yfSwapFlt10.append(yfSwpFlt10)
    yfSwapFix2.append(yfSwpFix2)
    yfSwapFix3.append(yfSwpFix3)
    yfSwapFix5.append(yfSwpFix5)
    yfSwapFix7.append(yfSwpFix7)
    yfSwapFix10.append(yfSwpFix10)

#####################
# CREATE DATAFRAMES #
#####################
#... deposits and futures
depoVal = pd.DataFrame(depoValue, index=fut_prices.index, columns=[libor_rates.columns[1]])
futuresVal = pd.DataFrame(futuresValue, index=fut_prices.index, columns=fut_prices.columns)
futuresEnd = pd.DataFrame(futuresEnd, index=fut_prices.index, columns=fut_prices.columns)
#...payment schedules for swaps
pmtSchedule2 = pd.DataFrame(swpPmtDates2, index=fut_prices.index,
                                     columns =[str(np.int(12*(n+1)/freqFlt)) + 'M' for n in range(2*freqFlt)])
pmtSchedule3 = pd.DataFrame(swpPmtDates3, index=fut_prices.index,
                                     columns =[str(np.int(12*(n+1)/freqFlt)) + 'M' for n in range(3*freqFlt)])
pmtSchedule5 = pd.DataFrame(swpPmtDates5, index=fut_prices.index,
                                     columns =[str(np.int(12*(n+1)/freqFlt)) + 'M' for n in range(5*freqFlt)])
pmtSchedule7 = pd.DataFrame(swpPmtDates7, index=fut_prices.index,
                                     columns =[str(np.int(12*(n+1)/freqFlt)) + 'M' for n in range(7*freqFlt)])
pmtSchedule10 = pd.DataFrame(swpPmtDates10, index=fut_prices.index,
                                      columns =[str(np.int(12*(n+1)/freqFlt)) + 'M' for n in range(10*freqFlt)])
#...year fractions for deposits and futures
yfDeposit = pd.DataFrame(yfDeposit, index=fut_prices.index, columns=[libor_rates.columns[1]])
yfFuturesVal = pd.DataFrame(yfFuturesVal, index=fut_prices.index, columns=fut_prices.columns)
yfFuturesEnd = pd.DataFrame(yfFuturesEnd, index=fut_prices.index, columns=fut_prices.columns)
#...year fractions for swaps
yfSwapFlt2 = pd.DataFrame(yfSwapFlt2, index=fut_prices.index,
                                   columns =[str(np.int(12*(n+1)/freqFlt)) + 'M' for n in range(2*freqFlt)])
yfSwapFlt3 = pd.DataFrame(yfSwapFlt3, index=fut_prices.index,
                                columns =[str(np.int(12*(n+1)/freqFlt)) + 'M' for n in range(3*freqFlt)])
yfSwapFlt5 = pd.DataFrame(yfSwapFlt5, index=fut_prices.index,
                                columns =[str(np.int(12*(n+1)/freqFlt)) + 'M' for n in range(5*freqFlt)])
yfSwapFlt7 = pd.DataFrame(yfSwapFlt7, index=fut_prices.index,
                                columns =[str(np.int(12*(n+1)/freqFlt)) + 'M' for n in range(7*freqFlt)])
yfSwapFlt10 = pd.DataFrame(yfSwapFlt10, index=fut_prices.index,
                                 columns =[str(np.int(12*(n+1)/freqFlt)) + 'M' for n in range(10*freqFlt)])
yfSwapFix2 = pd.DataFrame(yfSwapFix2, index=fut_prices.index,
                                columns =[str(np.int(12*(n+1)/freqFlt)) + 'M' for n in range(2*freqFlt)])
yfSwapFix3 = pd.DataFrame(yfSwapFix3, index=fut_prices.index,
                                columns =[str(np.int(12*(n+1)/freqFlt)) + 'M' for n in range(3*freqFlt)])
yfSwapFix5 = pd.DataFrame(yfSwapFix5, index=fut_prices.index,
                                columns =[str(np.int(12*(n+1)/freqFlt)) + 'M' for n in range(5*freqFlt)])
yfSwapFix7 = pd.DataFrame(yfSwapFix7, index=fut_prices.index,
                                columns =[str(np.int(12*(n+1)/freqFlt)) + 'M' for n in range(7*freqFlt)])
yfSwapFix10 = pd.DataFrame(yfSwapFix10, index=fut_prices.index,
                                 columns =[str(np.int(12*(n+1)/freqFlt)) + 'M' for n in range(10*freqFlt)])


# ### Convexity Adjustments

# In[ ]:


# DEFINE USER DEFINED FUNCTIONS
def calibrateHW2Swaptions(swaptionVolatilities,depositRates,swapMktRates,
                                volType=ql.Normal,meanRevLevel=0.05):
    '''
    Given swaption volatilities, depo rates and swap rates,
    calibrates the HW parameters.
    '''
    meanReversionMeanFixed = []
    volatilityMeanFixed = []

    if volType == ql.Normal:
        div = 10000
    else:
        div = 100

    for dt, row in swaptionVolatilities.iterrows():
        #...
        tradeDate = ql.Date(dt.day,dt.month,dt.year)
        settleDate = cal.advance(tradeDate, np.int(settleDays), ql.Days, conv)
        ql.Settings.instance().evaluationDate = tradeDate

        #...check if we have MeanFixed deposit data and swap data for the given date
        if (depositRates.index.isin([dt]).sum() > 0) & (swapMktRates.index.isin([dt]).sum() > 0):
            #...set up term structure: deposit rate helper
            depoDates = [ql.Period(3,ql.Months)]
            depoRates = [depositRates.loc[dt].values[0]]
            #...combine dates and rates to create helper object
            depoHelper = [ql.DepositRateHelper(ql.QuoteHandle(ql.SimpleQuote(rate/100.0)),
                             length, settleDays, cal, conv, True, dayCount)
                            for rate, length in zip(depoRates, depoDates)]
            #...set up term structure: swap rate helper object
            swapRatesTenor = [ql.Period(6,ql.Months), ql.Period(9,ql.Months), ql.Period(1,ql.Years),
                              ql.Period(18,ql.Months), ql.Period(2,ql.Years), ql.Period(3,ql.Years),
                              ql.Period(4,ql.Years), ql.Period(5,ql.Years), ql.Period(6,ql.Years),
                              ql.Period(7,ql.Years), ql.Period(8,ql.Years), ql.Period(9,ql.Years),
                              ql.Period(10,ql.Years)]
            swapRates = swapMktRates.loc[dt,:]
            #...create a boolean array that is true if data is available and false otherwise
            swapBool = (~swapRates.isna()).tolist()
            #...now filter the tenors and rates based on this boolean array
            swapRatesTenor = list(compress(swapRatesTenor, swapBool))
            swapRates = list(compress(swapRates, swapBool))
            swapRatesHelper = [ql.SwapRateHelper(ql.QuoteHandle(ql.SimpleQuote(rate/100)),
                              tenor, cal, fixLegTenorTS, conv, fixLegDayCount,
                                iborIndex) for rate, tenor in zip(swapRates,swapRatesTenor)]
            #...combine helper objects
            termStructureHelpers = depoHelper + swapRatesHelper
            #...set up yield curve using log linear discounting on discount factors
            termStructure = ql.PiecewiseLogLinearDiscount(tradeDate, termStructureHelpers, dayCount)
            termStructure.enableExtrapolation()
            #...set up term structure
            termStructure = ql.YieldTermStructureHandle(termStructure)
            index = ql.USDLibor(ql.Period(3,ql.Months), termStructure)

            # SET UP HULL WHITE CALIBRATION
            #..the swaption vols we will use for calibration
            mktVols = namedtuple('mktVols','start, tenor, vol')
            startYr = np.arange(1,10,1)
            tenorYr = np.arange(9,0, 1)
            calibrationData = [mktVols(np.int(startYr[idx]), np.int(tenorYr[idx]),
                                              vol/div) for idx, vol in enumerate(row) if ~np.isnan(vol)]
            try:
                #...set up Hull White Model
                model = ql.HullWhite(termStructure,meanRevLevel,0.01);
                engine = ql.JamshidianSwaptionEngine(model)
                #...create swaption helpers that we will use for the optimization
                #...(note that this works for normal vols, and not Black vols)
                swaptionHelpers = []
                for swaption in calibrationData:
                    volHandle = ql.QuoteHandle(ql.SimpleQuote(swaption.vol))
                    swaptionHelper = ql.SwaptionHelper(ql.Period(swaption.start, ql.Years),
                                                         ql.Period(swaption.tenor, ql.Years),
                                                         volHandle, index, fixLegTenorSH, fixLegDayCount,
                                                       fltLegDayCount, termStructure,
                                                       ql.BlackCalibrationHelper.RelativePriceError,
                                                       ql.nullDouble(), 1.0, volType,0.)
                    swaptionHelper.setPricingEngine(engine)
                    swaptionHelpers.append(swaptionHelper)
                #... fit theta and compute alpha and sigma that best matches market swaption vols
                optMethod = ql.LevenbergMarquardt(1.0e 8,1.0e 8,1.0e 8)
                endCrit = ql.EndCriteria(10000, 100, 1e 6, 1e 8, 1e 8)
                model.calibrate(swaptionHelpers, optMethod, endCrit, ql.NoConstraint(), [], [True, False])
                #... extract mean reversion and volatility from the calibrated model
                a, sigma = model.params()
            except:
                a = np.nan
                sigma = np.nan
        else:
            a = np.nan
            sigma = np.nan
        #...add parameters to lists
        meanReversionMeanFixed.append(a)
        volatilityMeanFixed.append(sigma)

    dfMeanFixed = pd.DataFrame([meanReversionMeanFixed, volatilityMeanFixed],
                                 index=['a','sigma'],columns=swaptionVolatilities.index).transpose()
    return dfMeanFixed;

def convexityBias(a,sigma,futPrices,yrFracDiff,yrFracSettle):
    '''
    Formulas are consistent with the ones reported in:
    http://www2.rotman.utoronto.ca/~hull/TechnicalNotes/TechnicalNote1.pdf
    And double checked with numerical results from the Ametrano & Bianchetti paper.
    Inputs:
    a: calibrated mean reversion parameter
    sigma: calibrated sigma parameter
    futPrices: futures prices
    yrFracDiff: difference in year fractions between end and value dates
    yrFracSettle: year fractions betweeen settle date and futures value date
    Outputs:
    rFutCvx: futures rates adjusted for convexity
    cvxBias: the convexity bias used
    '''
    # FORMAT VARIABLES
    #...find common indices between all variables and filter based on those
    commonIdx = reduce(np.intersect1d, (a.index, sigma.index, yrFracDiff.index, yrFracSettle.index, futPrices.index))
    a = a.loc[commonIdx]
    sigma = sigma.loc[commonIdx]
    futPrices = futPrices.loc[commonIdx]
    yrFracDiff = yrFracDiff.loc[commonIdx]
    yrFracSettle = yrFracSettle.loc[commonIdx]

    # COMPUTE THE CONVEXITY
    #...set up two variables
    B0T1 = (1  yrFracSettle.mul( a,axis=0).applymap(np.exp)).div(a,axis=0)
    BT1T2 = (1  yrFracDiff.mul( a,axis=0).applymap(np.exp)).div(a,axis=0)
    #...compute the convexity bias
    cvxBias = BT1T2.div(yrFracDiff).mul(B0T1.pow(2).mul(a,axis=0).mul(2).add((1 yrFracSettle.mul(a,axis=0).mul( 2).applymap(np.exp)).mul(BT1T2))).mul(sigma.pow(2).div(a.mul(4)),axis=0)
    #...convert futures rates to continuously compounded
    rFut = (100 futPrices)/100
    rFutCont = rFut.div(4).add(1).applymap(np.log).mul(4)
    #...compute convexity adjusted futures rate
    rFutContCvx = rFutCont  cvxBias
    #...convert rate to quarterly compounding
    rFutCvx = rFutContCvx.div(4).applymap(np.exp).sub(1).mul(4)
    return rFutCvx, cvxBias


# #### Hull White One Factor Model

# In[ ]:


```

```
if currency == 'USD':
    #...define number of days after trade that deal settles
    settleDays = np.int(2)
    #...specify calendar for market
    cal = ql.UnitedStates()
    #...specify day count convention
    dayCount = ql.Actual360()
    #...define frequency of floating payments and other date rules
    conv = ql.ModifiedFollowing
    #...
    iborIndex = ql.USDLibor(ql.Period(3, ql.Months))
    #...
    fixLegTenorTS = ql.Semiannual
    fixLegTenorSH = ql.Period(6,ql.Months)
    fixLegDayCount = ql.Thirty360()
    fltLegDayCount = ql.Actual360()

if currency == 'EUR':
    #...define number of days after trade that deal settles
    settleDays = np.int(2)
    #...specify calendar for market
    cal = ql.TARGET()
    #...specify day count convention
    dayCount = ql.Actual360()
    #...define frequency of floating payments and other date rules
    conv = ql.ModifiedFollowing
    #...
    iborIndex = ql.Euribor6M()
    #...
    fixLegTenorTS = ql.Annual
    fixLegTenorSH = ql.Period(6,ql.Months)
    fixLegDayCount = ql.Thirty360(ql.Thirty360.BondBasis)
    fltLegDayCount = ql.Actual360()

if ((currency=='USD') | (currency=='EUR')) & (adjustCvx==True):
    #...Hull White calibrated parameters for normal swaption
    volatilities
    hwNormCalParams = calibrateHW2Swaptions(swaption_norm,
    libor_rates, swap_rates_HW, ql.Normal, 0.05)
    if currency == 'USD':
        #...calibrate for lognormal volatilies if currency is USD
        hwBlackCalParams = calibrateHW2Swaptions(swaption_black,
    libor_rates, swap_rates_HW, ql.ShiftedLognormal, 0.05)
    else:
        hwBlackCalParams = pd.DataFrame()
    #...merge data into one dataframe and fill NaN's by linear interpolation
    hwCalParams = hwNormCalParams.combine_first(hwBlackCalParams)
    hwCalParams = hwCalParams.interpolate(method='linear', axis=0).bfill()
    #...compute the convexity bias using the calibrated parameters
    cvxFutRates, cvxBias = convexityBias(hwCalParams['a'], hwCalParams['sigma'],
                  fut_prices, yfFuturesEnd.sub(yfFuturesVal), yfFuturesVal)
    #...compute convexity adjusted futures prices
    cvxAdjFutPrices = 100 cvxFutRates*100


# ### Implied Swap Rates from Futures

# In[ ]:


# DEFINE USER DEFINED FUNCTIONS
def LogXInterpolation(yearFractions, discountFactors, interpolatedYearFractions,method):
    '''
    Inputs:
    yearFractions: known year fractions (x values), input as pandas Series
    discountFactors: known discount factors (y values), input as pandas Series
    interpolatedYearFractions: year fractions we want to find discount factors for, input as list
    Output:
    interpolatedDiscountFactors: list of interpolated discount factors at interpolatedYearFractions
    '''
    #...account for NaN's in inputs
    yearFractions.dropna(inplace=True)
    discountFactors.dropna(inplace=True)
    #...check if length of variables is equal
    yearFractions = yearFractions[:len(discountFactors)]
    #lenBool = (len(yearFractions) == len(discountFactors))
    #...check length of variables, if zero, we return NaN's
    if np.isnan(interpolatedYearFractions).sum() == 0:
        #...remove first STIR futures contract if it overlaps with deposit 3M
        if yearFractions[0]==yearFractions[1]:
            yearFractions.drop([fut_prices.columns[0]],axis=0,inplace=True)
            discountFactors.drop([fut_prices.columns[0]],axis=0,inplace=True)
        #...do log transformation on discount factors
        logDiscountFactors = np.log(discountFactors.astype(np.float64))
        #...use scipy functions to set up linear interpolation
        interpolatedCurve = interp1d(yearFractions, logDiscountFactors, kind=method, fill_value='extrapolate')
        #...extract interpolated discount factors
        interpolatedDiscountFactors = [np.exp(interpolatedCurve(dt)) for dt in interpolatedYearFractions]
    else:
        interpolatedDiscountFactors = list(np.ones(len(interpolatedYearFractions))*np.nan)
    #...return the interpolated values
    return interpolatedYearFractions, interpolatedDiscountFactors;

def LogXInterpolationDataFrame(yfSum, dfAct, swpPmt, method):
    '''
    Inputs:
    yfSum: Dataframe containing known year fractions
    dfAct: Dataframe containing known discount factors
    swpPmt: Dataframe containing year fractions we want to interpolate for
    Outputs:
    yfIntp: Dataframe containing interpolated year fractions.
    dfIntp: Dataframe containing interpolated discount factors.
    '''
    #...create an empty list to store
    yfIntpList = []
    dfIntpList = []
    #...make sure the three inputs have the same indices
    commonIdx = reduce(np.intersect1d, (yfSum.index, dfAct.index, swpPmt.index))
    yfSum = yfSum.loc[commonIdx]
    dfAct = dfAct.loc[commonIdx]
    swpPmt = swpPmt.loc[commonIdx]
    #...interpolate for the discount factors
    for dt, row in yfSum.iterrows():
        #...known year fractions
        yfActual = row.dropna()
        #...known discount factors
        dfActual = dfAct.loc[dt,:].dropna()
        #...year fractions we want to find through interpolation
        yfIntp = list(swpPmt.loc[dt,:].dropna())
        if (len(yfIntp) > 0) & (len(dfActual) > 0):
            #...run the function doing the log linear interpolation on the discount factors
            yfIntp, dfIntp = LogXInterpolation(yfActual, dfActual, yfIntp, method)
            #...append interpolated results to lists
            yfIntpList.append(yfIntp)
            dfIntpList.append(dfIntp)
        else:
            yfIntpList.append(list(np.ones(len(swpPmt.loc[dt,:]))*np.nan))
            dfIntpList.append(list(np.ones(len(swpPmt.loc[dt,:]))*np.nan))
    #...construct dataframes based on lists
    yfIntp = pd.DataFrame(yfIntpList, index=yfSum.index,
                          columns=[str(np.int(12*(n+1)/freqFlt)) + 'M' for n in range(swpPmt.shape[1])])
    dfIntp = pd.DataFrame(dfIntpList, index=yfSum.index,
                          columns=[str(np.int(12*(n+1)/freqFlt)) + 'M' for n in range(swpPmt.shape[1])])
    return yfIntp, dfIntp;

def pvFloatingLeg(dfsForward, dfsDiscount):
    '''
    Computes present value of floating leg
    in vanilla IRS given inputs:
    * dfsForward: discount factors from forward curve.
    * dfsDiscount: discount factors from discounting curve.
    '''
    pvFlt = pd.concat([1/dfsForward.iloc[:,0],dfsForward.shift(1,axis=1).div(dfsForward).dropna(axis=1,how='all')],
           axis=1).sub(1).mul(dfsDiscount).sum(axis=1,skipna=False)
    return pvFlt;

def pvFixedLeg(dfsFix, yearFracFix, freqFlt=freqFlt, freqFixed=freqFixed):
    '''
    Computes present value of fixed leg
    in vanilla IRS given inputs:
    * freqFlt: frequency of floating payments
    * freqFixed: frequency of fixed payments
    * yearFracFix: year fractions between dates in IRS date schedule
    that accounts for day count convention in fixed leg.
    * dfsFix: discount factors from discounting curve
    '''
    #...two numbers that help extract the fixed schedule later
    fixStart = np.int((freqFlt/freqFixed) 1)
    fixInterval = np.int(freqFlt/freqFixed)
    #...compute year fractions on fixed leg frequency
    yfFixFrac = pd.concat([yearFracFix.iloc[:,fixStart],
                           yearFracFix.iloc[:,fixStart::fixInterval].diff(axis=1).dropna(axis=1,how='all')],axis=1)
    #...compute the present value of the fixed leg
    pvFix = dfsFix.iloc[:,fixStart::fixInterval].mul(yfFixFrac).sum(axis=1)
    return pvFix;


# ##### Discount Factors

# In[ ]:


# COMPUTE OIS DISCOUNT FACTORS

# Source for this synthetic OIS rate methodology:
# OIS Discounting and Dual Curve Stripping Methodology, Quantitative Analytics, Bloomberg, Dec 2017
# If we are doing USD, we need to compute synthetic OIS rates for 7 and 10 year tenors
if currency == 'USD':
    #...extract ibor swap rates
    iborSR = ois_rates.loc[:,['USSWAP7 Curncy','USSWAP10 Curncy']].div(100)
    #...extract ois basis spreads
    oisBasis = ois_rates.loc[:,['USBG7 Curncy','USBG10 Curncy']].div(10000)
    #...compute r_q
    rq = iborSR.mul(360/365).div(2).add(1).pow(2/4).sub(1).mul(4)
    #...compute approximated OIS rate
    approxOIS = (1+((rq.values  oisBasis.values)/4))**4   1
    approxOIS = ((1+approxOIS/360)**90   1)*4
    ois_rates.iloc[:,[9,10]] = approxOIS*100
    ois_rates.drop(['USSWAP10 Curncy','USBG10 Curncy'],axis=1,inplace=True)

#...change names of columns
ois_rates = ois_rates.div(100)
```

```
colNames = ['OIS3M','OIS6M','OIS9M','OIS1Y','OIS18M','OIS2Y',
                       'OIS3Y','OIS4Y','OIS5Y','OIS7Y','OIS10Y']
colNames = [colNames[idx] for idx in range(ois_rates.shape[1])]
ois_rates.columns = colNames

#...compute year fractions
oisYearFractions = []
for dt, row in ois_rates.iterrows():
    #...convert date to QuantLib format
    tradeDate = ql.Date(dt.day,dt.month,dt.year)
    #...set settle date at T+2 days
    settleDate = cal.advance(tradeDate, np.int(settleDays), ql.Days, conv)
    #...compute maturity dates
    tenors = [3,6,9,12,18,24,36,48,60,84,120]
    tenors = [tenors[idx] for idx in range(ois_rates.shape[1])]
    maturityDates = [cal.advance(settleDate, np.int(tenor), ql.Months, conv) for tenor in tenors]
    yearFractions = [dayCount.yearFraction(settleDate,dt) for dt in maturityDates]
    #...add results to lists
    oisYearFractions.append(yearFractions)

colNames = ['OIS3M','OIS6M','OIS9M','OIS1Y','OIS18M','OIS2Y',
                       'OIS3Y','OIS4Y','OIS5Y','OIS7Y','OIS10Y']
colNames = [colNames[idx] for idx in range(ois_rates.shape[1])]
oisYrFrac = pd.DataFrame(oisYearFractions,index=ois_rates.index,columns=colNames)
oisYrFracDiff = pd.concat([oisYrFrac.iloc[:,0],oisYrFrac.diff(axis=1).dropna(how='all',axis=1)],axis=1)

#...compute discount factors up to and including one year
oisDfs = 1 / ois_rates.iloc[:,:4].mul(oisYrFrac.iloc[:,:4]).add(1)
#...compute discount factors over one year
for i in range(ois_rates.iloc[:,4:].shape[1]):
    i += 4
    up = 1   oisYrFracDiff.iloc[:,:i].mul(oisDfs).sum(axis=1).mul(ois_rates.iloc[:,i],axis=0)
    down = 1 + ois_rates.iloc[:,i].mul(oisYrFracDiff.iloc[:,i],axis=0)
    oisDfsAdd = up/down
    oisDfs = pd.concat([oisDfs,oisDfsAdd],axis=1)
oisDfs.columns = oisYrFrac.columns


```

```
# In[ ]:


# COMPUTE IBOR DISCOUNT FACTORS
#...combine deposit year fractions and futures year fractions
yrFrac = pd.concat([yfDeposit,pd.concat([yfDeposit,yfFuturesEnd],
                                                    axis=1).diff(axis=1).dropna(axis=1,how='all')],axis=1)
#...combine deposit rates and futures rates
rates = pd.concat([libor_rates.iloc[:, 1],(100 fut_prices)],axis=1).div(100)
#...compute discount factors
futDfs = (1 / rates.mul(yrFrac).add(1)).cumprod(axis=1)
#...we want to make entire row NaN if we do not have values for 3M deposit
futDfs.loc[futDfs.iloc[:,0].isna(),:] = np.nan
#...
yrFracSum = yrFrac.cumsum(axis=1)

if ((currency=='USD') | (currency=='EUR')) & (adjustCvx==True):
    # COMPUTE CONVEXITY ADJUSTED IBOR DISCOUNT FACTORS
    #...combine deposit rates and futures rates
    ratesCvx = pd.concat([libor_rates.iloc[:, 1],(100 cvxAdjFutPrices)],axis=1).div(100)
    #...compute discount factors
    futDfsCvx = (1 / ratesCvx.mul(yrFrac).add(1)).cumprod(axis=1)
    #...we want to make entire row NaN if we do not have values for 3M deposit
    futDfsCvx.loc[futDfsCvx.iloc[:,0].isna(),:] = np.nan
    #...drop rows where we don't have sufficient data
    futDfsCvx = futDfsCvx.loc[(~futDfsCvx.isna()).sum(axis=1)>9]


# ##### Interpolation

# In[ ]:


# LOG CUBIC INTERPOLATION ON OIS DISCOUNT FACTORS
#...compute interpolated discount factors for the 2 year swap
yfFltOIS2, dfFltOIS2 = LogXInterpolationDataFrame(oisYrFrac, oisDfs, yfSwapFlt2, 'cubic')
yfFixOIS2, dfFixOIS2 = LogXInterpolationDataFrame(oisYrFrac, oisDfs, yfSwapFix2, 'cubic')
#...compute interpolated discount factors for the 3 year swap
```

```
yfFltOIS3, dfFltOIS3 = LogXInterpolationDataFrame(oisYrFrac, oisDfs, yfSwapFlt3, 'cubic')
yfFixOIS3, dfFixOIS3 = LogXInterpolationDataFrame(oisYrFrac, oisDfs, yfSwapFix3, 'cubic')
#...compute interpolated discount factors for the 5 year swap
yfFltOIS5, dfFltOIS5 = LogXInterpolationDataFrame(oisYrFrac, oisDfs, yfSwapFlt5, 'cubic')
yfFixOIS5, dfFixOIS5 = LogXInterpolationDataFrame(oisYrFrac, oisDfs, yfSwapFix5, 'cubic')
#...compute interpolated discount factors for the 7 year swap
yfFltOIS7, dfFltOIS7 = LogXInterpolationDataFrame(oisYrFrac, oisDfs, yfSwapFlt7, 'cubic')
yfFixOIS7, dfFixOIS7 = LogXInterpolationDataFrame(oisYrFrac, oisDfs, yfSwapFix7, 'cubic')
#...compute interpolated discount factors for the 10 year swap
yfFltOIS10, dfFltOIS10 = LogXInterpolationDataFrame(oisYrFrac, oisDfs, yfSwapFlt10, 'cubic')
yfFixOIS10, dfFixOIS10 = LogXInterpolationDataFrame(oisYrFrac, oisDfs, yfSwapFix10, 'cubic')


# In[ ]:


# LOG LINEAR INTERPOLATION ON IBOR DISCOUNT FACTORS
##################################################
# IBOR DISCOUNT FACTORS UNADJUSTED FOR CONVEXITY #
##################################################
#...compute interpolated discount factors for the 2 year swap
yfFlt2, dfFlt2 = LogXInterpolationDataFrame(yrFracSum, futDfs, yfSwapFlt2, 'linear')
yfFix2, dfFix2 = LogXInterpolationDataFrame(yrFracSum, futDfs, yfSwapFix2, 'linear')
#...compute interpolated discount factors for the 3 year swap
yfFlt3, dfFlt3 = LogXInterpolationDataFrame(yrFracSum, futDfs, yfSwapFlt3, 'linear')
yfFix3, dfFix3 = LogXInterpolationDataFrame(yrFracSum, futDfs, yfSwapFix3, 'linear')
#...compute interpolated discount factors for the 5 year swap
yfFlt5, dfFlt5 = LogXInterpolationDataFrame(yrFracSum, futDfs, yfSwapFlt5, 'linear')
yfFix5, dfFix5 = LogXInterpolationDataFrame(yrFracSum, futDfs, yfSwapFix5, 'linear')
#...compute interpolated discount factors for the 7 year swap
yfFlt7, dfFlt7 = LogXInterpolationDataFrame(yrFracSum, futDfs,
    yfSwapFlt7, 'linear')
yfFix7, dfFix7 = LogXInterpolationDataFrame(yrFracSum, futDfs,
    yfSwapFix7, 'linear')
#...compute interpolated discount factors for the 10 year swap
yfFlt10, dfFlt10 = LogXInterpolationDataFrame(yrFracSum, futDfs,
    yfSwapFlt10, 'linear')
yfFix10, dfFix10 = LogXInterpolationDataFrame(yrFracSum, futDfs,
    yfSwapFix10, 'linear')

if ((currency=='USD') | (currency=='EUR')) & (adjustCvx==True):
    ################################################
    # IBOR DISCOUNT FACTORS ADJUSTED FOR CONVEXITY #
    ################################################
    #...compute interpolated discount factors for the 2 year swap
    yfFltCvx2, dfFltCvx2 = LogXInterpolationDataFrame(yrFracSum.loc[
    futDfsCvx.index],
                                                              futDfsCvx,
    yfSwapFlt2.loc[futDfsCvx.index], 'linear')
    yfFixCvx2, dfFixCvx2 = LogXInterpolationDataFrame(yrFracSum.loc[
    futDfsCvx.index],
                                                              futDfsCvx,
    yfSwapFix2.loc[futDfsCvx.index], 'linear')
    #...compute interpolated discount factors for the 3 year swap
    yfFltCvx3, dfFltCvx3 = LogXInterpolationDataFrame(yrFracSum.loc[
    futDfsCvx.index],
                                                              futDfsCvx,
    yfSwapFlt3.loc[futDfsCvx.index], 'linear')
    yfFixCvx3, dfFixCvx3 = LogXInterpolationDataFrame(yrFracSum.loc[
    futDfsCvx.index],
                                                              futDfsCvx,
    yfSwapFix3.loc[futDfsCvx.index], 'linear')
    #...compute interpolated discount factors for the 5 year swap
    yfFltCvx5, dfFltCvx5 = LogXInterpolationDataFrame(yrFracSum.loc[
    futDfsCvx.index],
                                                              futDfsCvx,
    yfSwapFlt5.loc[futDfsCvx.index], 'linear')
    yfFixCvx5, dfFixCvx5 = LogXInterpolationDataFrame(yrFracSum.loc[
    futDfsCvx.index],
                                                              futDfsCvx,
    yfSwapFix5.loc[futDfsCvx.index], 'linear')
    #...compute interpolated discount factors for the 7 year swap
    yfFltCvx7, dfFltCvx7 = LogXInterpolationDataFrame(yrFracSum.loc[
    futDfsCvx.index],
                                                              futDfsCvx,
    yfSwapFlt7.loc[futDfsCvx.index], 'linear')
    yfFixCvx7, dfFixCvx7 = LogXInterpolationDataFrame(yrFracSum.loc[futDfsCvx.index],
                                                                              futDfsCvx, yfSwapFix7.loc[futDfsCvx.index], 'linear')
    #...compute interpolated discount factors for the 10 year swap
    yfFltCvx10, dfFltCvx10 = LogXInterpolationDataFrame(yrFracSum.loc[futDfsCvx.index],
                                                                                futDfsCvx, yfSwapFlt10.loc[futDfsCvx.index], 'linear')
    yfFixCvx10, dfFixCvx10 = LogXInterpolationDataFrame(yrFracSum.loc[futDfsCvx.index],
                                                                                futDfsCvx, yfSwapFix10.loc[futDfsCvx.index], 'linear')


# ##### Futures Implied Swap Rates

# In[ ]:


# ####################################################################################################
# FUTURES IMPLIED SWAP RATES UNADJUSTED FOR CONVEXITY USING IBOR DISCOUNTING #
# ####################################################################################################
#...present values of floating legs
pvFlt2 = pvFloatingLeg(dfFlt2, dfFlt2)
pvFlt3 = pvFloatingLeg(dfFlt3, dfFlt3)
pvFlt5 = pvFloatingLeg(dfFlt5, dfFlt5)
pvFlt7 = pvFloatingLeg(dfFlt7, dfFlt7)
pvFlt10 = pvFloatingLeg(dfFlt10, dfFlt10)
#...present values of fixed legs
pvFix2 = pvFixedLeg(dfFix2, yfFix2)
pvFix3 = pvFixedLeg(dfFix3, yfFix3)
pvFix5 = pvFixedLeg(dfFix5, yfFix5)
pvFix7 = pvFixedLeg(dfFix7, yfFix7)
pvFix10 = pvFixedLeg(dfFix10, yfFix10)
#...implied 2 year swap rate
futSwp2 = pvFlt2/pvFix2
#...implied 3 year swap rate
futSwp3 = pvFlt3/pvFix3
#...implied 5 year swap rate
```

```
futSwp5 = pvFlt5/pvFix5
#...implied 7 year swap rate
futSwp7 = pvFlt7/pvFix7
#...implied 10 year swap rate
futSwp10 = pvFlt10/pvFix10
#...store results in a dataframe
futSwp = pd.concat([futSwp2,futSwp3,futSwp5,futSwp7,futSwp10],axis=1)
futSwp.columns = ['2Yr','3Yr','5Yr','7Yr','10Yr']

# ################################################################################
# FUTURES IMPLIED SWAP RATES ADJUSTED FOR CONVEXITY USING IBOR DISCOUNTING #
# ################################################################################
if ((currency=='USD') | (currency=='EUR')) & (adjustCvx==True):
    #...present values of floating legs
    pvFltCvx2 = pvFloatingLeg(dfFltCvx2, dfFltCvx2)
    pvFltCvx3 = pvFloatingLeg(dfFltCvx3, dfFltCvx3)
    pvFltCvx5 = pvFloatingLeg(dfFltCvx5, dfFltCvx5)
    pvFltCvx7 = pvFloatingLeg(dfFltCvx7, dfFltCvx7)
    pvFltCvx10 = pvFloatingLeg(dfFltCvx10, dfFltCvx10)
    #...present values of fixed legs
    pvFixCvx2 = pvFixedLeg(dfFixCvx2, yfFix2)
    pvFixCvx3 = pvFixedLeg(dfFixCvx3, yfFix3)
    pvFixCvx5 = pvFixedLeg(dfFixCvx5, yfFix5)
    pvFixCvx7 = pvFixedLeg(dfFixCvx7, yfFix7)
    pvFixCvx10 = pvFixedLeg(dfFixCvx10, yfFix10)
    #...implied 2 year swap rate
    futSwpCvx2 = pvFltCvx2/pvFixCvx2
    #...implied 3 year swap rate
    futSwpCvx3 = pvFltCvx3/pvFixCvx3
    #...implied 5 year swap rate
    futSwpCvx5 = pvFltCvx5/pvFixCvx5
    #...implied 7 year swap rate
    futSwpCvx7 = pvFltCvx7/pvFixCvx7
    #...implied 10 year swap rate
    futSwpCvx10 = pvFltCvx10/pvFixCvx10
    #...store results in a dataframe
    futSwpCvx = pd.concat([futSwpCvx2,futSwpCvx3,futSwpCvx5, futSwpCvx7,futSwpCvx10],axis=1)
    futSwpCvx.columns = ['2Yr','3Yr','5Yr','7Yr','10Yr']

```

```
# ############################################################################
# FUTURES IMPLIED SWAP RATES UNADJUSTED FOR CONVEXITY USING OIS DISCOUNTING #
# ############################################################################
#...present values of floating legs
pvFltOIS2 = pvFloatingLeg(dfFlt2, dfFltOIS2)
pvFltOIS3 = pvFloatingLeg(dfFlt3, dfFltOIS3)
pvFltOIS5 = pvFloatingLeg(dfFlt5, dfFltOIS5)
pvFltOIS7 = pvFloatingLeg(dfFlt7, dfFltOIS7)
pvFltOIS10 = pvFloatingLeg(dfFlt10, dfFltOIS10)
#...present values of fixed legs
pvFixOIS2 = pvFixedLeg(dfFixOIS2, yfFixOIS2)
pvFixOIS3 = pvFixedLeg(dfFixOIS3, yfFixOIS3)
pvFixOIS5 = pvFixedLeg(dfFixOIS5, yfFixOIS5)
pvFixOIS7 = pvFixedLeg(dfFixOIS7, yfFixOIS7)
pvFixOIS10 = pvFixedLeg(dfFixOIS10, yfFixOIS10)
#...implied 2 year swap rate
futSwpOIS2 = pvFltOIS2/pvFixOIS2
#...implied 3 year swap rate
futSwpOIS3 = pvFltOIS3/pvFixOIS3
#...implied 5 year swap rate
futSwpOIS5 = pvFltOIS5/pvFixOIS5
#...implied 7 year swap rate
futSwpOIS7 = pvFltOIS7/pvFixOIS7
#...implied 10 year swap rate
futSwpOIS10 = pvFltOIS10/pvFixOIS10
#...store results in a dataframe
futSwpOIS = pd.concat([futSwpOIS2,futSwpOIS3,futSwpOIS5,futSwpOIS7,
    futSwpOIS10],axis=1)
futSwpOIS.columns = ['2Yr','3Yr','5Yr','7Yr','10Yr']

# ###########################################################################
# FUTURES IMPLIED SWAP RATES ADJUSTED FOR CONVEXITY USING OIS DISCOUNTING #
# ###########################################################################

if ((currency=='USD') | (currency=='EUR')) & (adjustCvx==True):
    #...present values of floating legs
    pvFltOISCvx2 = pvFloatingLeg(dfFltCvx2, dfFltOIS2)
    pvFltOISCvx3 = pvFloatingLeg(dfFltCvx3, dfFltOIS3)
    pvFltOISCvx5 = pvFloatingLeg(dfFltCvx5, dfFltOIS5)
    pvFltOISCvx7 = pvFloatingLeg(dfFltCvx7, dfFltOIS7)
    pvFltOISCvx10 = pvFloatingLeg(dfFltCvx10, dfFltOIS10)
    #...present values of fixed legs will be same as before
    #...implied 2 year swap rate
    futSwpOISCvx2 = pvFltOISCvx2/pvFixOIS2
    #...implied 3 year swap rate
    futSwpOISCvx3 = pvFltOISCvx3/pvFixOIS3
    #...implied 5 year swap rate
    futSwpOISCvx5 = pvFltOISCvx5/pvFixOIS5
    #...implied 7 year swap rate
    futSwpOISCvx7 = pvFltOISCvx7/pvFixOIS7
    #...implied 10 year swap rate
    futSwpOISCvx10 = pvFltOISCvx10/pvFixOIS10
    #...store results in a dataframe
    futSwpOISCvx = pd.concat([futSwpOISCvx2,futSwpOISCvx3,futSwpOISCvx5,futSwpOISCvx7,futSwpOISCvx10],axis=1)
    futSwpOISCvx.columns = ['2Yr','3Yr','5Yr','7Yr','10Yr']


# ### Export Results to Excel

# In[ ]:


# FORMAT THE MARKET SWAP RATES
mktSwp = swap_rates.div(100)
mktSwp.columns = ['2Yr','3Yr','5Yr','7Yr','10Yr']

# CREATE DATAFRAME WITH IMM DATES
#...find third Wednesday in each month
startDate = fut_prices.index[0]
endDate = fut_prices.index[ 1]
thirdWednesday = ql.Schedule(ql.Date(startDate.day,startDate.month,startDate.year),
                                               ql.Date(endDate.day,endDate.month,endDate.year),
                                               ql.Period(ql.Monthly),cal,conv,conv,ql.DateGeneration.ThirdWednesday,False)
thirdWednesday = [datetime(dt.year(), dt.month(), dt.dayOfMonth()) for dt in list(thirdWednesday)]
#...store dates in dataframe (monthly and quarterly)
immDts = list(fut_prices.index[[ql.IMM.isIMMdate(ql.Date(dt.day,dt.
    month,dt.year)) for dt in fut_prices.index]])
immDts = pd.DataFrame([immDts,thirdWednesday]).transpose()

###########################
# EXPORT RESULTS TO EXCEL #
###########################
writer = pd.ExcelWriter(excelFilename, engine='xlsxwriter')
mktSwp.to_excel(writer, sheet_name='mktSwp')
futSwp.to_excel(writer, sheet_name='futSwp')
futSwpOIS.to_excel(writer, sheet_name='futSwpOIS')
if ((currency=='USD') | (currency=='EUR')) & (adjustCvx==True):
    futSwpCvx.to_excel(writer, sheet_name='futSwpCvx')
    futSwpOISCvx.to_excel(writer, sheet_name='futSwpOISCvx')
immDts.to_excel(writer, sheet_name='immDts')
writer.save()
```