Eötvös Loránd University

Faculty of Science

Zita Flóra Brückler

Pricing cross-currency products

MSc Thesis

Supervisor:

Dr. Gábor Molnár - Sáska

Department of Probability Theory and Statistics

Budapest, 2014

Acknowledgements

I would like to express my thanks to Gábor Molnár - Sáska for his valuable and constructive suggestions during the planning and development of this thesis. I am really thankful to Ferenc Izsák for his help and advice on the numerical PDE solving, and Balázs Kovács for helping in the first steps.
Many thanks to Alexandra Harkai and Ákos Gyarmati for their help in getting familiar with the 'FX-world' and PRDCs.
I wish to thank János Papp for his help in MATLAB.

I am really grateful to my family and friends for their support.

Zita Flóra Brückler

Contents

1 Introduction

Since the financial crisis, the valuation of financial products has reached a high level of interest. In the world of financial mathematics more and more sensitive and precise models were constructed. Also the investors tend to look for long-dated products, which are not so effected by the short-term market changes. It is important to have a model which incorporates all relevant patterns from the market (see [12]).

The **Power-Reverse Dual-Currency** is a complex FX product which is commonly traded in the Japanese financial market. It is long-dated, pays an FX-linked coupon in exchange for a floating payment. Its sensitivity to volatility smiles or skews makes the product more interesting from modeling aspects.

In this thesis we are focusing on the pricing of the above mentioned PRDC products. In the second chapter we give a general overview of this product, have a look on the Japanese economical background which led to the popularity of PRDCs, and on the development of this security, i.e. how the exotic features appeared in this population.

In the third section a cross-currency model with a local volatility function is introduced, based on the paper of Vladimir V. Piterbarg from 2005. In this article Piterbarg took a strong emphasis on how the volatility skew can be represented, and how the calibration of the model parameters can be done easily. In this thesis we also walk through on these topics.

We can say, that the Partial Differential Equation (PDE) pricing framework is not so commonly used. In the fourth section we give a numerical methodology on solving the PDE, which comes from the model described previously for pricing such securities like PRDCs. Of course, before using a model for pricing, the issue of calibration is also needed to be discussed.

To summarize, in this thesis we study the cross-currency model with local volatility function from [3]: how it can be used for the valuation of a PRDC; how the calibration problem is easier in this framework and finally how it can be implemented in practice.

Let me note that, for all topics mentioned later on in these pages it is worth taking more and more deeper investigation. This area of financial mathematics - like all the others as well- is always under development.

2 PRDC – Power Reverse Dual Currency swap

Power Reverse Dual Currencies (PRDCs) are cross-currency exotics widely traded by Japanese investors. In this chapter we will look through briefly the Japanese economical factors which led to the circumstances where PRDCs could satisfy Japanese traders looking for higher yield. Also, the main properties of this exotic financial product will be described.[1]

2.1 The economical background

If someone takes a look at interest rates in Japanese yen (at least prior to 2008), they can recognize how low were they, especially versus the US or Australian dollar. Japan has faced nearly zero interest rates for much of the first decade of the 21st century. This can be seen in Figure 1.[2]

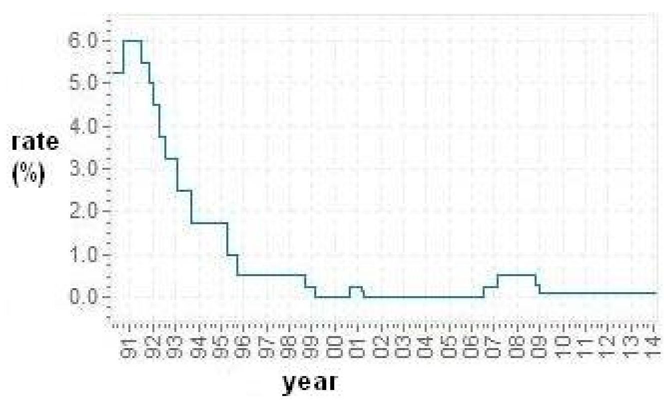

Figure 1: Bank of Japan uncollateralized overnight call rate

From 1998 to 2008 even the 5-year swap rates remained below 2%, compared to the US rates which showed a trend around 4% - 5% (except a short period in 2003).
The question speaks for itself: why not borrow in a currency where interest rates are low (in this case JPY), and convert it to such a currency where we can expect high interest (like USD or AUD)? This is what the **Carry Trade** is about.

[1]This section is based on [1] - Chapter 10.
[2]Source: *www.global-rates.com*

In a Carry Trade investors are betting against the appreciation of the Japanese yen. As we have seen it above, the yen rates are below the dollar rates, and because of no-arbitrage arguments, we expect forward USD/JPY to be lower than the spot USD/JPY. Let's see a simple example. Suppose that the spot USD/JPY FX-rate is at 100, yen interest rates are 2% and dollar interest rates are 5%. (Data is from 2007.) With these rates, if I borrow 100 yen, the cost is $100 \times (1 + 0.02)^5$. If I buy dollars for the 100 yen and invest it, the dollar investment is $1 \times (1 + 0.05)^5$. We want to determine the forward FX-rate (FX_{fwd}), so the yen value of the investment is $FX_{fwd} \times (1 + 0.05)^5$. From no-arbitrage, the investment and the borrowing need to give the same result, i.e.

$$100 \times (1 + 0.02)^5 = FX_{fwd} \times (1 + 0.05)^5 \quad \Rightarrow \quad FX_{fwd} = 86.5$$

Someone, who invests in a Carry Trade is betting against the fall of the USD/JPY FX-rate, expecting it to remain at current levels, i.e. they are betting that forwards will not be realized.

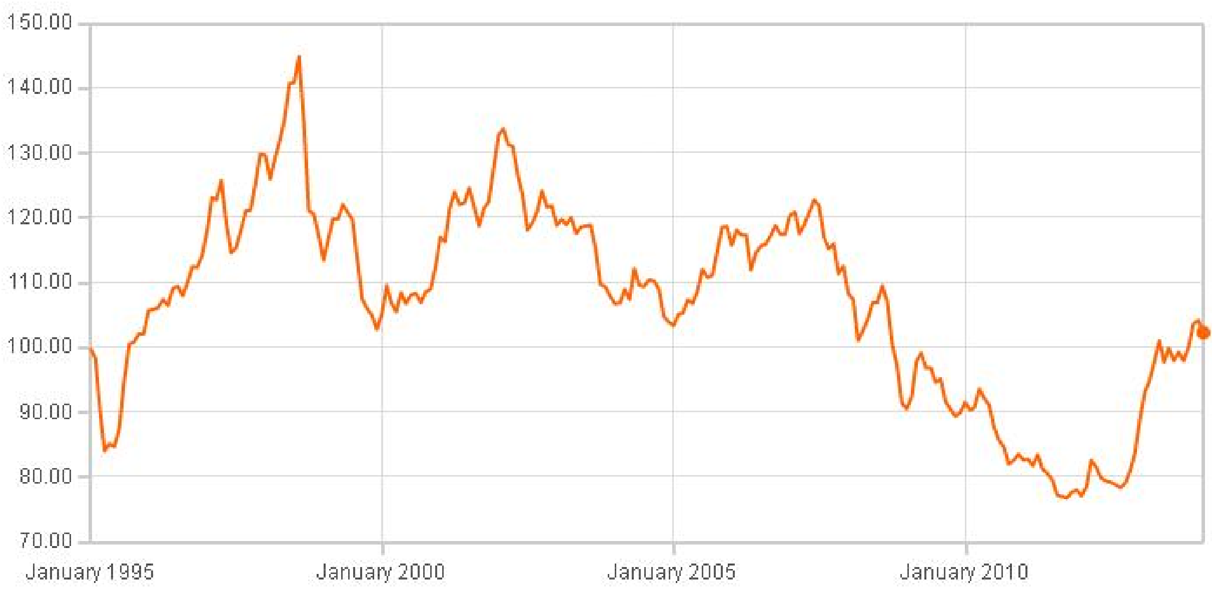

Figure 2: Value of spot USD/JPY from 1995 to 2014

Looking at the graph of the USD/JPY rate in the period 1996 to 2007 (Figure 2), we can see that it remained between 100 and 135.[3] One reason of this behavior of the spot

[3]Source: *http://www.oanda.com/currency/historical-rates/*

USD/JPY FX-rate is that Japanese government is trying to prevent yen appreciation to avoid hurting exporters.

To see the whole picture, we need to mention, that Carry Trades are not risk-free to invest in. If yen gets appreciated significantly, this leads to unwind the Carry Trades, which means selling US dollars and buying back Japanese yen. The effect will appear in the USD/JPY FX-rate as well, as it will show a dramatic fall. This happened from September 2008, when the circumstances of the global economy made people opt for safety. The USD/JPY dropped under 90, having the unwinding of the Carry Trades on the list of the causes. A similar situation was faced regarding the Australian dollar.

PRDCs have maturity like 20 - 30 years, and the long-dated foreign exchange market had the opportunity for its development, as most investors rather believe that USD/JPY will not follow what the forward rate suggests, even in 30 years from now.

2.2 About PRDCs in general

PRDCs first appeared in the market in 1995, and their rise is highly related to the above mentioned economical background in Japan. Investors are hungry for higher yield, and a coupon from a 30-year bond, with Japanese rates being so low, is not so attractive. The idea of having coupons linked to the spot FX-rate holds out a promise to gain more. The name – Power Reverse Dual Currency – suggests, that the first PRDC notes were a leveraged version of the common reverse dual note, i.e. an investment denominated in yen, while paying coupons in another, higher-yielding currency.

PRDC swaps (investigating from the issuer's side) pay FX-linked coupons (the so called PRDC coupons) in exchange for LIBOR floating-rate payments. The **simple PRDC coupon** is nothing more, but a call option on the FX-rate, having the payoff at a coupon date t:

$$L \times \max(S(t) - K, 0) \times DCF,$$

where $S(t)$ is the spot FX-rate at time t, K is strike and L is a multiplier. DCF is the accrual multiplier, i.e. if the coupons are not accrued annually, they are multiplied by the appropriate Day Count Fraction. For the sake of simplicity, we assume for the rest of this chapter, that coupons are paid annually, so DCF equals to 1.

With a typical choice of the parameters, when $L = 0.0016; K = 80$, and if USD/JPY remains at 115, the investor can reach a coupon level of 5.6%, which is much more favorable than a 3% annual coupon of a bond. Of course, if the FX-rate drops below 80, than the investor gets nothing, but they are convinced, that if it ever was under 80, it would not happen in a short time.

The more common coupon, which is the so called **PRDC coupon** has the following structure:

$$usdcpn \times \frac{S(t)}{K} - jpycpn,$$

where usdcpn and jpycpn are the foreign and domestic coupons, and K is often called the initial FX-rate.

As PRDC notes have gone through a development from highly exotic to most commonly traded and liquid product in the Japanese market, variations of some extra features are added to the basic PRDC structure to have it fitted more to the taste of the investors. A fixed rate period at the beginning of a trade guarantees a quite large fixed coupon (e.g. 6-8%) for the first couple of periods. This is about to compensate less preferential payoff parameters. Other examples is to floor (e.g. at 0.01%) and cap (typically at the same level of the fixed coupon) the FX-linked payoff. It is worth mentioning, that the PRDC coupon is always floored at 0.0%, the extra feature is when the floor value is above zero. Floor (if it is above zero) protects against the situation when the investor is stucked with having zero coupons, and cap gives a protection for the issuer. With these two parameters (flr, cap), the payoff has the form of

$$\min\left(\max\left(usdcpn \times \frac{S(t)}{K} - jpycpn, flr\right), cap\right).$$

If we assume, that $flr = 0$; $cap = +\infty$ (which are widely used settings), the PRDC coupon can be written still in the form of a call option on the FX-rate:

$$h \max(S(t) - k, 0), \quad \text{where } k = K\frac{jpycpn}{usdcpn}, \quad h = \frac{usdcpn}{K}.$$

2.2.1 Early Termination

To ensure protection for the issuer, and also to obtain higher yields for the investor, in most cases there is an embedded early termination. Without this, the terms of the

PRDC coupon would look much worse for the investor (for example strike at higher level). These features are not applied immediately from the effective date, but after the first couple of periods.

In **Callable PRDCs**, there are Bermudan callable structures, where the issuer has the right to call the trade. It is reasonable, when USD/JPY tends to stay high for long, as the attractive high coupons cannot be payed forever. **KnockOut PRDCs** have a trigger level – KO barrier –, and if the USD/JPY FX-rate reaches or get above the KO barrier the whole structure get cancelled. As the forward FX-rate is decreasing while the PRDC is getting closer to maturity, the typical trigger level also steps down for example in every year. Callables have been traded from 1997, KnockOut PRDCs have been available since 1999. **TARN PRDCs** represent such a feature, that the structure is cancelled when the total coupon (the sum of the PRDC coupons in the history of the trade) exceeds a TARN level. This TARN level is usually between 15% - 40%. All of the above three have the danger for the investor, that if USD/JPY drops, and a floor higher than zero is not set, then they face the situation of receiving zero coupons. We note, that many combinations exist, like Callable TARN PRDC, Callable KnockOut PRDC.

2.2.2 Redemption Strike

There is an other extra appearing on PRDC trades, called the redemption strike. If a redemption strike R is given, the investor receives $\frac{1}{R}$ on the termination date (T) in foreign currency, instead of its yen principal. To express this in yen, its value is $\frac{S(T)}{R}$, where $S(T)$ is the spot FX-rate at expiry. Investors can lose their principal protection, as this fractional can happen to be less than 1. In the market, trades usually both have redemption strike and early termination feature as well, however if early termination occurs, the redemption does not have effect, it is considered only when the trade survives until termination. As Japanese investors do believe that early termination will occur, or if it will not, then the FX-rate will not drop under the redemption strike, they are happy to risk their principal protection for being compensated with a higher initial coupon or a higher coupon multiplier. On the other hand, we need to mention, that the forward FX-rate goes below the value of R between about the $15^{th} - 20^{th}$ year of the trade. So again, the bet is against the expectation of what is suggested by the forward rate.

2.3 Chooser PRDCs

A more complicated version of PRDCs described above, is the **Chooser PRDC**. Investors are really hungry for high yields and declining margins also have led to newer financial products. The name is confusing, as not the investor is who can choose, but the issuer. A Chooser PRDC pays the minimum of USD/JPY and AUD/JPY PRDC coupons. They also come with redemption strikes. If the product does not terminate early, the minimum amount of applying the dollar or Australian dollar redemption strikes is paid, instead of the yen notional. TARN and KnockOut features are represented for them as well, however the dimensionality of the product makes it non-conducive to PDE pricing and this makes dealers not to apply callability to it. The other question of the pricing is the correlation between AUD/JPY and USD/JPY. A high correlation between them increases the price of the product for the issuer, as a rise in one of them will not be followed by a fall in the other. In this thesis, because of the mentioned reasons, the pricing of Chooser PRDCs is not in scope, but it would be a really interesting and challenging project for future work.

3 A cross-currency model with FX volatility skew

We have seen in the previous chapter, that there is a continuing interest in PRDC swaps. This leads to the requirement of a sophisticated model and numerical valuation of cross-currency interest rate derivatives. The valuation of them using a PDE (Partial Differential Equation) method is not well-developed, the popular approach among financial institutes for pricing PRDC swaps is Monte-Carlo simulation, however the slow convergence is just one of the known disadvantages of this methodology.

Foreign exchange (FX) interest rate hybrids face the effect of the movement of the spot FX rate and also the interest rates in both currencies. The most common modeling of such products is a three-factor modeling framework, which consists of a one-factor log-normal model for the spot FX rate, and the interest rates of the two currencies are driven by one-factor Gaussian models.

3.1. Definition. ***(Gaussian class)*** *A short term interest rate model is said to belong to the Gaussian class if it can be written as the following linear differential equation*

$$\mathrm{d}r(t) = \mu_r(t, r(t))\mathrm{d}t + \sigma_r(t, r(t))\mathrm{d}W(t) = (\mu_1(t)r(t) + \mu_2(t))\mathrm{d}t + \sigma_2(t)\mathrm{d}W(t).$$

3.1. Note. *The well-known lognormal model has the relationship with the Gaussian, that a short term interest rate model is lognormal* $\Longleftrightarrow$ $\ln r(t)$ *is Gaussian.*

This way the number of factors is kept to the minimum (a total of three), and a very efficient calibration to the at-the-money options on the FX rate can be used.
On the other side, the lognormality of the FX rate do not allow us to model an important behavior which exists in reality, namely that FX options exhibit a significant volatility skew. This cannot be well captured by the log-normal distribution. Moreover, cross-currency derivatives with exotic features mentioned earlier, are really sensitive to the FX volatility skew. Therefore a model, which incorporates the FX skew is needed. By using stochastic volatility we would introduce a new stochastic factor, so for the sake of holding the speed and accuracy of the calibration, staying in the context of local volatility seems to be a good choice.

V. V. Piterbarg in his paper [3] has built a cross-currency model that incorporates FX volatility skew by using a local volatility function.

3.1 The model

As we have a cross-currency model, there must be a 'domestic' and a 'foreign' currency considered (signed as d, f in indices). In our case, Japanese yen is the domestic and US dollar is the foreign currency.

Let

- $\mathbf{P}$ be the domestic risk-neutral measure;
- $P_i(t,T), \quad i = d, f$ be the prices of the domestic and foreign zero-coupon discount bonds (in their respective currencies)[4];
- $r_i(t), \quad i = d, f$ be the short rates in the domestic and foreign currencies;
- $S(t)$ be the spot FX-rate, expressed in the units of domestic currency per one unit of the foreign currency.

The model is the following:

$$\begin{aligned} \mathrm{d}P_d(t,T)/P_d(t,T) &= r_d(t)\mathrm{d}t + \sigma_d(t,T)\mathrm{d}W_d(t), \\ \mathrm{d}P_f(t,T)/P_f(t,T) &= r_f(t)\mathrm{d}t - \rho_{fS}\sigma_f(t,T)\gamma(t,S(t))\mathrm{d}t + \sigma_f(t,T)\mathrm{d}W_f(t), \\ \mathrm{d}S(t)/S(t) &= (r_d(t) - r_f(t))\mathrm{d}t + \gamma(t,S(t))\mathrm{d}W_S(t), \end{aligned} \tag{3.1}$$

where $(W_d(t),\ W_f(t),\ W_S(t))$ is a Brownian motion under $\mathbf{P}$, having the correlation matrix as

$$\begin{pmatrix} 1 & \rho_{df} & \rho_{dS} \\ \rho_{df} & 1 & \rho_{fS} \\ \rho_{dS} & \rho_{fS} & 1 \end{pmatrix}.$$

3.2. Note. *There is a 'quanto' drift adjustment for* $\mathrm{d}P_f(t,T)$*, which comes from changing the measure from the foreign risk-neutral to the domestic risk-neutral (see [5]). Knowing that* $e^{-r_d(t)t}S(t)P_f(t,T)$ *is a martingale under* $\mathbf{P}$*, the drift of* $\dfrac{\mathrm{d}(e^{-r_d(t)t}S(t)P_f(t,T))}{e^{-r_d(t)t}S(t)P_f(t,T)}$

[4]The risk-free bond is used, as the probability of bankruptcy is not considered in the short rates.

should be zero.

Assuming

$$\frac{\mathrm{d}P_f(t,T)}{P_f(t,T)} = \mu_f(t)\mathrm{d}t + \sigma_f(t,T)\mathrm{d}W_f(t),$$

we want to find μ_f*.*

$$\begin{aligned}\frac{\mathrm{d}(e^{-r_d(t)t}S(t)P_f(t,T))}{e^{-r_d(t)t}S(t)P_f(t,T)} &= -r_d(t)\mathrm{d}t + \frac{\mathrm{d}S(t)}{S(t)} + \frac{\mathrm{d}P_f(t,T)}{P_f(t,T)} + \frac{\mathrm{d}\left[S(t),P_f(t,T)\right]_t}{S(t)P_f(t,T)} = \\ &= -r_d(t)\mathrm{d}t + (r_d(t)-r_f(t))\mathrm{d}t + \gamma(t,S(t))\mathrm{d}W_S(t) + \mu_f(t)\mathrm{d}t + \\ &\quad + \sigma_f(t,T)\mathrm{d}W_f + \gamma(t,S(t))\sigma_f(t,T)\rho_{fS}\mathrm{d}t\end{aligned}$$

From 'drift $=0$*' we get*

$$r_d(t) - r_f(t) + \mu_f(t) + \gamma(t,S(t))\sigma_f(t,T)\rho_{fS} = r_d(t).$$

Therefore

$$\mu_f(t) = r_f(t) - \gamma(t,S(t))\sigma_f(t,T)\rho_{fS}.$$

Continuing the model description, we have

$$\sigma_i(t,T) = \sigma_i(t)\int_t^T e^{-\int_t^s \varkappa_i(u)\mathrm{d}u}\mathrm{d}s, \quad i=d,f, \tag{3.2}$$

where $\sigma_d(t),\ \sigma_f(t),\ \varkappa_d(t),\ \varkappa_f(t)$ are deterministic functions.

The local volatility function $\gamma(t,x)$ imposes the FX volatility skew on the model. It is common choosing this function independent from x, but for the stability of calibration the following parametrization is used,

$$\gamma(t,x) = \nu(t)\left(\frac{x}{L(t)}\right)^{\beta(t)-1}, \tag{3.3}$$

where $\nu(t)$ is the 'relative volatility function', $\beta(t)$ is a time-dependent constant elasticity of variance (CEV) parameter and $L(t)$ is a time-dependent scaling constant (see [3]).

The forward FX rate ($F(t,T)$) can be obtained from the spot FX-rate and the interest rates in the two currencies by the well-known formula, coming from no-arbitrage arguments:

$$F(t,T) = \frac{P_f(t,T)}{P_d(t,T)}S(t). \tag{3.4}$$

3.2 The PDE for valuation

For the short rates in a one-factor model a Markovian representation holds true. Let's assume the extended Vasicek model:

$$\mathrm{d}r_i(t) = (\theta_i(t) - \varkappa_i(t) r_i(t))\mathrm{d}t + \sigma_i(t)\mathrm{d}W_i(t), \quad i = d, f, \tag{3.5}$$

under the respective risk-forward measures (it uses a bond with maturity T as numeraire). The closed forms for $\theta_i(t)$ are the following (see [11]):

$$\theta_i(t) = \frac{\partial f(0,t)}{\partial T} + \varkappa_i f(0,t) + \frac{\sigma_i^2}{2\varkappa_i}\left(1 - e^{-2\varkappa_i t}\right) \quad i = d, f. \tag{3.6}$$

As zero-coupon bonds arise from the short rates via deterministic functions, the model (3.1) gives us a Markovian representation in three variables $(r_d(.),\ r_f(.),\ S(.))$. The price of any product whose payoff is a function of the FX-rate and the interest rates of the two currencies must satisfy a PDE described in the following theorem (see also [6]).

3.1. Theorem. *Let $V = V(t, r_d, r_f, S)$ denote the value of a security (such as a PRDC) at time t, with a terminal payoff measurable to the σ- algebra at maturity time. Assume, that $V \in \mathcal{C}^{1,2}$ on $[T_{start}, T_{end}) \times \mathbb{R}_+^3$.*
Then V satisfies the PDE

$$\begin{aligned} & V_t + (\theta_d(t) - \varkappa_d(t) r_d) V_{r_d} + \\ & +(\theta_f(t) - \rho_{fS}\sigma_f(t)\gamma(t, S(t)) - \varkappa_f(t) r_f) V_{r_f} + \\ & +(r_d - r_f) S V_S + \\ & +\tfrac{1}{2}\sigma_d^2(t) V_{r_d r_d} + \tfrac{1}{2}\sigma_f^2(t) V_{r_f r_f} + \tfrac{1}{2}\gamma^2(t,S) S^2 V_{SS} + \\ & +\rho_{df}\sigma_d(t)\sigma_f(t) V_{r_d r_f} + \rho_{dS}\sigma_d(t)\gamma(t,S) S V_{r_d S} + \rho_{fS}\sigma_f(t)\gamma(t,S) S V_{r_f S} = \\ & = r_d V. \end{aligned} \tag{3.7}$$

Proof: The dynamics of the foreign interest rate r_f will be the following, after changing the measure to the domestic one:

$$\mathrm{d}r_f(t) = (\theta_f(t) - \varkappa_f(t) r_f(t) - \rho_{fS}\sigma_f(t)\gamma(t, S(t)))\mathrm{d}t + \sigma_f(t)\mathrm{d}W_f(t).$$

If we look at the normalized price process of any security, i.e. the discounted value by the bond price, they are martingales. This holds for V, and since it is an Itô process, the drift must be zero. Calculating the drift using the Itô formula, and setting it to zero we get the following.

$$\begin{aligned}
\mathrm{d}\frac{V(s,r_d,r_f,t)}{B(t)} &= \frac{1}{B(t)}\mathrm{d}V + V(-r_d)\frac{1}{B(t)}\mathrm{d}t = \\
&= \frac{1}{B(t)}\Bigg(V_t\mathrm{d}t + V_{r_d}\mathrm{d}r_d + V_{r_f}\mathrm{d}r_f + V_S\mathrm{d}S + \frac{1}{2}\gamma^2(t,S(t))S^2V_{SS}+ \\
&\quad +\frac{1}{2}\sigma_d^2(t)V_{r_dr_d} + \frac{1}{2}\sigma_f^2(t)V_{r_fr_f} + \frac{1}{2}2\rho_{dS}\sigma_d(t)\gamma(t,S(t))SV_{Sr_d}+ \\
&\quad +\frac{1}{2}2\rho_{fS}\sigma_f(t)\gamma(t,S(t))SV_{Sr_f} + \frac{1}{2}2\rho_{df}\sigma_d(t)\sigma_f(t)V_{r_dr_f} - r_dV\Bigg),
\end{aligned}$$

where $B(t)$ denotes the domestic bond, i.e. $\mathrm{d}B(t) = r_d(t)B(t)\mathrm{d}t$.

If we substitute the dynamics of the interest rates and the FX rate, collect the 'dt' part, set it equal to zero, then finally multiply the equation with $B(t)$, we got exactly the PDE (3.7).

□

3.3 Pricing options on the FX rate

As it was mentioned above, we expect the volatility skew to be imposed by the local volatility function of the FX-rate (3.3). The main question is how this function can be calibrated. Normally, a volatility is calibrated to the prices of options on the underlying asset. Options on the FX rate are traded with a lot of maturity and strike, and it is impossible to choose one maturity and strike to be relevant for our security (in the case of PRDCs). Moreover, PRDCs with exotic features cannot be decomposed into simple FX options. All in all, the volatility function needs to be calibrated to prices of all available FX options.

A call option with strike K and maturity T pays $(S(T)-K)^+$ at T, and at time 0 its value equals to

$$c(T,K) = \mathbf{E}_0\left(e^{-\int_0^T r_d(s)\mathrm{d}s}(S(T)-K)^+\right).$$

The dynamics of the spot FX rate are quite complex, but the forward FX rate is a martingale under the domestic forward measure. So it is convenient to rewrite $c(T,K)$ in the terms of the forward FX rate, and study the dynamics of $F(t,T)$ under the domestic T-forward measure $\mathbf{P}^T$ (for which $P_d(.,T)$ is a numeraire. Then, the value of an FX option is

$$c(T,K) = P_d(0,T)\mathbf{E}_0^T\left((F(T,T)-K)^+\right),$$

and for the dynamics the following holds true.

3.2. Theorem. *The forward FX rate $F(t,T)$ has the dynamics*

$$\mathrm{d}F(t,T)/F(t,T) = \sigma_f(t,T)\mathrm{d}W_f^T(t) - \sigma_d(t,T)\mathrm{d}W_d^T + \gamma(t,F(t,T)D(t,T))\mathrm{d}W_S^T(t), \quad (3.8)$$

where $(W_d^T(t),\ W_f^T(t),\ W_S^T(t))$ is a Brownian motion under the domestic T-forward measure $\mathbf{P}^T$. What is more, there exists a Brownian motion $W_F(t)$ under $\mathbf{P}^T$, such that

$$\frac{\mathrm{d}F(t,T)}{F(t,T)} = \Lambda(t,F(t,T)D(t,T))\mathrm{d}W_F(t), \quad (3.9)$$

where

$$\begin{aligned}\Lambda(t,x) &= (a(t)+b(t)\gamma(t,x)+\gamma^2(t,x))^{\frac{1}{2}},\\ a(t) &= (\sigma_f(t,T))^2+(\sigma_d(t,T))^2-2\rho_{df}\sigma_f(t,T)\sigma_d(t,T),\\ b(t) &= 2\rho_{fS}\sigma_f(t,T)-2\rho_{dS}\sigma_d(t,T),\end{aligned}$$

and

$$D(t,T) \stackrel{d}{=} \frac{P_d(t,T)}{P_f(t,T)}.$$

Proof: To apply Itô's lemma to (3.4), we need the following dynamics (also computed with Itø's lemma):

$$\begin{aligned}\mathrm{d}\frac{1}{P_d(t,T)} &= -\frac{1}{P_d(t,T)}\left[(r_d(t)+\sigma_d^2(t,T))\mathrm{d}t+\sigma_d(t,T)\mathrm{d}W_d(t)\right],\\ \mathrm{d}\frac{P_f(t,T)}{P_d(t,T)} &= \frac{P_f(t,T)}{P_d(t,T)}\Bigg(r_f(t)\mathrm{d}t-\rho_{fS}\sigma_f(t,T)\gamma(t,S(t))\mathrm{d}t+\sigma_f(t,T)\mathrm{d}W_f(t)-\\ &\quad -(r_d\mathrm{d}t+\sigma_d(t,T)\mathrm{d}W_d(t))-\sigma_d^2(t,T)\mathrm{d}t-\\ &\quad -\frac{1}{2}2\sigma_d(t,T)\sigma_f(t,T)\rho_{fd}\mathrm{d}t+\frac{1}{2}2\sigma_d^2(t,T)\mathrm{d}t\Bigg) =\end{aligned}$$

$$
\begin{aligned}
= \; & \frac{P_f(t,T)}{P_d(t,T)}\left[(r_f(t)-r_d(t)-\rho_{fS}\sigma_f(t,T)\gamma(t,S(t))-\sigma_d(t,T)\sigma_f(t.T)\rho_{fd})\mathrm{d}t+\right.\\
& \left.+\sigma_f(t,T)\mathrm{d}W_f(t)-\sigma_d\mathrm{d}W_d(t)\right].
\end{aligned}
$$

Now we can calculate the dynamics of $F(t,T)$:

$$
\begin{aligned}
\mathrm{d}F(t,T) &= \mathrm{d}\left(\frac{P_f(t,T)}{P_d(t,T)}S(t)\right) = \\
&= \frac{P_f(t,T)}{P_d(t,T)}\mathrm{d}S(t)+S(t)\mathrm{d}\frac{P_f(t,T)}{P_d(t,T)}+\mathrm{d}\left[\frac{P_f(t,T)}{P_d(t,T)},S(t)\right]_t = \\
&= F(t,T)\big[(r_d(t)-r_f(t))\mathrm{d}t+\gamma(t,S(t))\mathrm{d}W_S(t)+\\
&\quad +(r_f(t)-r_d(t)-\rho_{fS}\sigma_f(t,T)\gamma(t,S(t))-\sigma_d(t,T)\sigma_f(t,T)\rho_{fd})\mathrm{d}t+\\
&\quad +\sigma_f(t,T)\mathrm{d}W_f(t)-\sigma_d\mathrm{d}W_d(t)+\\
&\quad +(\sigma_f(t,T)\gamma(t,S(t))\rho_{fS}-\sigma_d(t,T)\gamma(t,S(t))\rho_{dS})\mathrm{d}t\big]. \quad \Rightarrow
\end{aligned}
$$

$$
\begin{aligned}
\frac{\mathrm{d}F(t,T)}{F(t,T)} &= \sigma_f(t,T)\mathrm{d}W_f(t)-\sigma_d(t,T)\gamma(t,S(t))\rho_{dS}\mathrm{d}t-\\
&\quad -\sigma_d(t,T)\mathrm{d}W_d(t)+\\
&\quad +\gamma(t,S(t))\mathrm{d}W_S(t)-\sigma_d(t,T)\sigma_f(t,T)\rho_{fd}\mathrm{d}t.
\end{aligned}
$$

Using that $F(t,T)$ is a martingale under $\mathbf{P}^T$, and defining $W_f^T(t)$, $W_d^T(t)$, $W_S^T(t)$ the following way

$$
\begin{aligned}
\mathrm{d}W_f^T(t) &= \mathrm{d}W_f(t)-\frac{\sigma_d(t,T)\gamma(t,S(t))\rho_{dS}}{\sigma_f(t,T)}\mathrm{d}t,\\
\mathrm{d}W_d^T(t) &= \mathrm{d}W_d(t),\\
\mathrm{d}W_S^T(t) &= \mathrm{d}W_S(t)-\frac{\sigma_d(t,T)\sigma_f(t,T)\rho_{fd}}{\gamma(t,S(t))}\mathrm{d}t,
\end{aligned}
$$

(3.8) is proven. Let's denote $\mathrm{d}W_F$ as

$$
\mathrm{d}W_F=\frac{1}{\Lambda(t,F(t,T)D(t,T))}\left(\sigma_f(t,T)\mathrm{d}W_f^T(t)-\sigma_d(t,T)\mathrm{d}W_d^T+\gamma(t,F(t,T)D(t,T))\mathrm{d}W_S^T(t)\right), \tag{3.10}
$$

so the equation (3.9) comes from (3.8). Then the only thing need to be proven is that W_F is a Brownian motion under $\mathbf{P}^T$, and this can be seen by computing the quadratic variation of $\mathrm{d}W_F$ from (3.10), as it gives $[W_F]_t=t$, from the Lévy-theorem the result is as expected.

□

3.3. Note. *If $\gamma(t,x)$ had been chosen to be independent from x, then we would have also got a deterministic function for the diffusion coefficient in (3.8). For this parametrization $F(T,T)$ is lognormally distributed.*

3.3.1 Simplifying the dynamics of the forward FX rate

In the general case, the term distribution of $F(.,T)$ is not so easy to identify. The diffusion coefficient of $\mathrm{d}F/F$ not just depends on F, but also on an other stochastic variable $D(t,T)$. The first step in simplifying the dynamics of the forward FX rate, is to write the SDE in a form that contains only F as stochastic variable.
V. V. Piterbarg in [3] derived an autonomous representation of the forward FX rate process that is exact for European options. Firstly, extension of the set of options being considered is introduced, by having $T > 0$, the settlement date of the forward FX rate, to be fixed, and including expiries before T.

$$c(t,T,K) = P_d(0,t)\mathbf{E}_0^T\left((F(t,T)-K)^+\right) \tag{3.11}$$

The task is to find the local volatility function $\tilde{\Lambda}(t,x)$, and it is motivated by Dupire's approach, as the values of options $\{c(t,T,K)\}$ in the model (3.1) are considered to be given, and $\tilde{\Lambda}(t,x)$ needs to be determined from them. Consider the model

$$\frac{\mathrm{d}F(t,T)}{F(t,T)} = \tilde{\Lambda}(t,F(t,T))\mathrm{d}W_F(t). \tag{3.12}$$

With this the values of European options $\{c(t,T,K)\}$, for all $0 < t < T,\ 0 \le K < \infty$, should match exactly the values of the same options from the original model. The following theorem gives us the solution (see [3]). The SDE which comes from $\tilde{\Lambda}(t,x)$ is called the 'Markovian representation' of the original one.

3.3. Theorem. *The local volatility function $\tilde{\Lambda}(t,x)$ for which the values of all European options $\{c(t,T,K)\}_{t,K}$ in the model (3.12) are the same as in the model (3.1) is given by*

$$\tilde{\Lambda}^2(t,x) = \mathbf{E}_0^T\left(\Lambda^2(t,F(t,T)D(t,T))\middle|F(t,T)=x\right).$$

Proof: We have the definition of $c(t,T,K)$ in (3.11). Using **Dupire's formula** (see [7]) to get the local volatility $\tilde{\Lambda}(t,x)$, for which the values of European options in the model

(3.12) match $\{c(t,T,K)\}_{t,K}$, we get

$$\left(K\tilde{\Lambda}(t,K)\right)^2 = 2\frac{\frac{\partial}{\partial t}\frac{c(t,T,K)}{P_d(0,t)}}{\frac{\partial^2}{\partial K^2}\frac{c(t,T,K)}{P_d(0,t)}} \tag{3.13}$$

Now we have to compute the right hand side of the above equation. For the first step

$$\mathrm{d}(F(t,T)-K)^+ = \chi_{\{F(t,T)>K\}}\mathrm{d}F(t,T) + \frac{1}{2}\delta_{\{F(t,T)=K\}}\mathrm{d}[F(t,T)].$$

We get the following from the fact, that $F(t,T)$ is a martingale under $\mathbf{P}^T$:

$$\mathbf{E}^T(F(t,T)-K)^+ - (F(0,T)-K)^+ = \frac{1}{2}\int_0^t \mathbf{E}^T\left(\delta_{\{F(t,T)=K\}}\mathrm{d}[F(t,T)]\right).$$

It is obvious, that

$$\mathbf{E}^T\left(\delta_{\{F(t,T)=K\}}\mathrm{d}[F(t,T)]\right) = \mathbf{E}^T\left(\delta_{\{F(t,T)=K\}}\right)\cdot\mathbf{E}^T\left(\mathrm{d}[F(t,T)]\middle|F(t,T)=K\right),$$

and that

$$\mathbf{E}^T\left(\delta_{\{F(t,T)=K\}}\right) = \frac{\partial^2}{\partial K^2}\mathbf{E}^T(F(t,T)-K)^+ = \frac{\partial^2}{\partial K^2}\frac{c(t,T,K)}{P_d(0,t)}.$$

From Theorem 3.2.

$$\mathrm{d}[F(t,T)] = F^2(t,T)\Lambda^2(t,F(t,T)D(t,T))\mathrm{d}t$$

is given, so

$$\mathbf{E}^T\left(\delta_{\{F(t,T)=K\}}\mathrm{d}[F(t,T)]\right) = \frac{\partial^2}{\partial K^2}\frac{c(t,T,K)}{P_d(0,t)}\cdot K^2\cdot\mathbf{E}^T\left(\Lambda^2(t,F(t,T)D(t,T))\middle|F(t,T)=K\right)\mathrm{d}t.$$

In particular,

$$\begin{aligned}\frac{\partial}{\partial t}\frac{c(t,T,K)}{P_d(0,t)} &= \frac{\partial}{\partial t}\left(\mathbf{E}^T(F(t,T)-K)^+ - (F(0,T)-K)^+\right) = \\ &= \frac{1}{2}\cdot\frac{\partial^2}{\partial K^2}\frac{c(t,T,K)}{P_d(0,t)}\cdot K^2\cdot\mathbf{E}^T\left(\Lambda^2(t,F(t,T)D(t,T))\middle|F(t,T)=K\right).\end{aligned}$$

Substitute this equality to (3.13),

$$\tilde{\Lambda}^2(t,K) = \mathbf{E}^T\left(\Lambda^2(t,F(t,T)D(t,T))\middle|F(t,T)=K\right),$$

and the theorem is proved.

□

3.1. Corollary. *For the purposes of European option pricing, the dynamics of the forward FX rate $F(.,T)$ under the measure $\mathbf{P}^T$ are* ***approximately*** *given by*

$$\frac{\mathrm{d}F(t,T)}{F(t,T)} = \hat{\Lambda}(t, F(t,T))\mathrm{d}W_F(t), \tag{3.14}$$

where

$$\begin{aligned}
\hat{\Lambda}(t,x) &= (a(t) + b(t)\hat{\gamma}(t,x) + \hat{\gamma}^2(t,x))^{\frac{1}{2}}, \\
\hat{\gamma}(t,x) &= \nu(t)\left(x\frac{D_0(t,T)}{L(t)}\right)^{\beta(t)-1}\left(1 + (\beta(t)-1)r(t)\left(\frac{x}{F(0,T)} - 1\right)\right), \\
D_0(t,T) &= \frac{P_d(0,t,T)}{P_f(0,t,T)},
\end{aligned}$$

where $P_i(s,t,T),\ i = d, f$ stand for the forward prices of the corresponding zero-coupon discount bonds,

and

$$r(t) = \frac{\int_0^t \chi_{Z,F}(s)\mathrm{d}s}{\int_0^t \chi_{F,F}(s)\mathrm{d}s},$$

where

$$\begin{aligned}
\chi_{Z,F}(t) &= -a(t) - \frac{b(t)}{2}\gamma(t, F(0,T)D_0(t,T)), \\
\chi_{F,F}(t) &= a(t) + b(t)\gamma(t, F(0,T)D_0(t,T)) + \gamma^2(t, F(0,T)D_0(t,T)).
\end{aligned} \tag{3.15}$$

For getting the corollary we need the following lemma.

3.1. Lemma. *For any $c \in \mathbb{R}$*

$$\mathbf{E}^T((D(t,T))^c \big| F(t,T) = x) \approx (D_0(t,T))^c\left(1 + c \cdot \frac{\int_0^t \chi_{Z,F}(s)\mathrm{d}s}{\int_0^t \chi_{F,F}(s)\mathrm{d}s} \cdot \left(\frac{x}{F(0,T)} - 1\right)\right),$$

where the definition of $\chi_{Z,F}(t),\ \chi_{F,F}(t)$ is in (3.15).

The lemma is proven in [3].

We know from Theorem 3.3., that for purposes of computing European option values, the dynamics of the forward FX rate follow

$$\frac{\mathrm{d}F(t,T)}{F(t,T)} = \tilde{\Lambda}(t, F(t,T))\mathrm{d}W_F(t),$$

where

$$\tilde{\Lambda}^2(t,x) = \mathbf{E}_0^T\left(\Lambda^2(t, F(t,T)D(t,T))\big|F(t,T)=x\right).$$

From Theorem 3.2.

$$\Lambda^2(t,x) = a(t) + b(t)\gamma(t,x) + \gamma^2(t,x).$$

Then

$$\tilde{\Lambda}^2(t,x) = a(t) + b(t)\mathbf{E}^T\left(\gamma(t, F(t,T)D(t,T))\big|F(t,T)=x\right) + \mathbf{E}^T\left(\gamma^2(t, F(t,T)D(t,T))\big|F(t,T)=x\right).$$

Define

$$\tilde{\gamma}(t,x) \stackrel{d}{=} \mathbf{E}^T\left(\gamma(t, F(t,T)D(t,T))\big|F(t,T)=x\right),$$

and approximate

$$\mathbf{E}^T\left(\gamma^2(t, F(t,T)D(t,T))\big|F(t,T)=x\right) \approx \tilde{\gamma}^2(t,x),$$

then

$$\tilde{\Lambda}^2(t,x) \approx a(t) + b(t)\tilde{\gamma}(t,x) + \tilde{\gamma}^2(t,x).$$

We have, from the form (3.3) of $\gamma(t,x)$, that

$$\tilde{\gamma}(t,x) = \nu(t)\left(\frac{F(t,T)}{L(t)}\right)^{\beta(t)-1}\mathbf{E}^T\left((D(t,T))^{\beta(t)-1}\Big|F(t,T)=x\right).$$

Denoting

$$\begin{aligned}
\hat{\gamma}(t,x) &= \nu(t)\left(\frac{x}{L(t)}\right)^{\beta(t)-1}(D_0(t,T))^{\beta(t)-1}\left(1 + (\beta(t)-1)\cdot\frac{\int_0^t \chi_{Z,F}(s)\mathrm{d}s}{\int_0^t \chi_{F,F}(s)\mathrm{d}s}\cdot\left(\frac{x}{F(0,T)}-1\right)\right),\\
\hat{\Lambda}^2(t,x) &= a(t) + b(t)\hat{\gamma}(t,x) + \hat{\gamma}^2(t,x),
\end{aligned}$$

and applying Lemma 3.1. with $c = \beta(t) - 1$, we get the Corollary 3.1.

Corollary 3.1. gives us an autonomous equation for the forward FX rate. It is a one-dimensional SDE, and the diffusion coefficient is given by a local volatility function $\hat{\Lambda}(t, x)$.

Moving forward in simplifying the dynamics of the forward FX rate, secondly we apply the so called 'skew averaging' technic (see also [3], [15]). With the approach of 'parameter averaging', time-dependent parameters are replaced by time-independent ones, which are called 'effective' parameters. This way we are able to directly relate the model and market parameters without any option calculations performed. The next step is based on the following generic result (see [3]).

3.2. Definition. *For the below theorem let us define* ***well approximated*** *in the following way: Let us define*

$$g_\varepsilon(t, x) = g\left(t\varepsilon^2, x_0 + (x - x_0)\varepsilon\right).$$

And with this definition

$$\begin{aligned}
\mathrm{d}X_\varepsilon(t) &= g_\varepsilon(t, X_\varepsilon(t))\mathrm{d}W(t), \\
\mathrm{d}X_\varepsilon(t) &= \sigma(t)\bar{g}_\varepsilon(Y_\varepsilon(t))\mathrm{d}W(t), \\
X_\varepsilon(0) &= X_0, \\
Y_\varepsilon(0) &= X_0.
\end{aligned}$$

Then

$$\mathbf{E}\left(X_\varepsilon(T) - X_0\right)^2 - \mathbf{E}\left(Y_\varepsilon(T) - X_0\right)^2 = o\left(\varepsilon^2\right).$$

3.4. Theorem. *Let $X(t)$ be a stochastic process defined by*

$$\mathrm{d}X(t) = g(t, X(t))\mathrm{d}W(t), \quad X(0) = X_0.$$

Then the distribution of $X(T)$ is ***well-approximated*** *by the distribution of $Y(T)$, where the stochastic process $Y(T)$ is defined by*

$$\mathrm{d}Y(t) = \sigma(t)\bar{g}(Y(t))\mathrm{d}W(t), \quad Y(0) = X_0.$$

The functions $\sigma(t)$, $\bar{g}(y)$ are the following

$$\begin{aligned}
\sigma(t) &= g(t, X_0), \\
\bar{g}(X_0) &= 1, \\
\frac{\partial}{\partial x}\bar{g}(x)\bigg|_{x=X_0} &= \int_0^T w(t)\frac{\frac{\partial}{\partial x}g(t,x)\big|_{x=X_0}}{g(t,X_0)}\mathrm{d}t,
\end{aligned}$$

and the weights $w(t)$ are given by

$$\begin{aligned}
w(t) &= \frac{u(t)}{\int_0^T u(t)\mathrm{d}t}, \\
u(t) &= g^2(t, X_0)\int_0^t g^2(s, X_0)\mathrm{d}s.
\end{aligned}$$

This theorem says that a time-dependent local volatility function $g(t,x)$ can be replaced with a time-constant one $\bar{g}(x)$, which has the slope at $x = X_0$ as a weighted average of time-dependent slopes of $g(t,x)$ with given weights.

One can apply this theorem to the forward FX rate from (3.14), then the result is the following.

3.5. Theorem. *To value options on the FX rate with maturity T, the forward FX rate can be approximated by the following SDE*

$$\mathrm{d}F(t,T) = \hat{\Lambda}(t, F(0,T))(\delta_F F(t,T) + (1-\delta_F)F(0,T))\mathrm{d}W_F(t), \tag{3.16}$$

where,

$$\delta_F = 1 + \int_0^T w(t)\frac{b(t)\eta(t) + 2\hat{\gamma}(t,F(0,T))\eta(t)}{2\hat{\Lambda}^2(t,F(0,T))}\mathrm{d}t, \tag{3.17}$$

$$\begin{aligned}
\eta(t) &= \hat{\gamma}(t, F(0,T))(1 + r(t))(\beta(t) - 1), \\
w(t) &= \frac{u(t)}{\int_0^T u(t)\mathrm{d}t}, \\
u(t) &= \hat{\Lambda}^2(t, F(0,T))\int_0^t \hat{\Lambda}^2(s, F(0,T))\mathrm{d}s.
\end{aligned}$$

In particular, $F(.,T)$ follows a 'standard displaced-diffusion SDE' with the skew parameter δ_F. The value of a call option on the FX rate with maturity T and strike K is

$$c(T,K) = P_d(0,T)c_{Black}\left(\frac{F(0,T)}{\delta_F},\ K + \frac{1-\delta_F}{\delta_F}F(0,T),\ \sigma_F\delta_F,\ T\right), \tag{3.18}$$

$$\sigma_F = \left(\frac{1}{T}\int_0^T \hat{\Lambda}^2(t, F(0,T))\mathrm{d}t\right)^{\frac{1}{2}}, \tag{3.19}$$

where $c_{Black}(F,K,\sigma,T)$ is the Black formula value for a call option with forward F, strike K, volatility σ and time to maturity T.

3.4. Note. *The Black formula is similar to the Black - Scholes formula except that the spot price of the underlying is replaced by a discounted forward price F.*

$$\begin{aligned} c_{Black}(F,K,\sigma,T) &= e^{-rT}\left[F\Phi(d_1) - K\Phi(d_2)\right], \\ \\ d_1 &= \frac{\log(F/K) + (\sigma^2/2)T}{\sigma\sqrt{T}}, \\ d_2 &= d_1 - \sigma\sqrt{T}. \end{aligned}$$

Proof: Here, we only prove the equations (3.16), (3.17). We need to apply Theorem 3.4. to

$$\begin{aligned} g(t,x) &= x\hat{\Lambda}(t,x), \\ \bar{g}(x) &= \delta_F\frac{x}{F(0,T)} + (1-\delta_F), \\ X_0 &= F(0,T). \end{aligned}$$

Then

$$\begin{aligned} \left.\frac{\partial}{\partial x}\bar{g}(x)\right|_{x=X_0} &= \frac{\delta_F}{F(0,T)}, \\ \frac{\partial}{\partial x}g(t,x) &= \hat{\Lambda}(t,x) + x\frac{\partial}{\partial x}\hat{\Lambda}(t,x) = \\ \\ &= \hat{\Lambda}(t,x) + x\frac{\partial}{\partial x}\left(\left(a(t) + b(t)\hat{\gamma}(t,x) + \hat{\gamma}^2(t,x)\right)^{\frac{1}{2}}\right) = \\ \\ &= \hat{\Lambda}(t,x) + x\frac{b(t)\frac{\partial}{\partial x}\hat{\gamma}(t,x) + 2\hat{\gamma}(t,x)\frac{\partial}{\partial x}\hat{\gamma}(t,x)}{2\left(a(t) + b(t)\hat{\gamma}(t,x) + \hat{\gamma}^2(t,x)\right)^{\frac{1}{2}}}. \end{aligned}$$

From these, we get

$$\left.\frac{\frac{\partial}{\partial x}g(t,x)}{g(t,x)}\right|_{x=X_0} = \frac{1}{F(0,T)} + \frac{b(t)\eta(t) + 2\hat{\gamma}(t,F(0,T))\eta(t)}{2F(0,T)\hat{\Lambda}^2(t,F(0,T))},$$

where

$$\eta(t) = F(0,T)\left.\frac{\partial}{\partial x}\hat{\gamma}(t,x)\right|_{x=F(0,T)}.$$

From Theorem 3.4. we have

$$\frac{\delta_F}{F(0,T)} = \int_0^T \left(\frac{1}{F(0,T)} + \frac{b(t)\eta(t) + 2\hat{\gamma}(t,F(0,T))\eta(t)}{2F(0,T)\hat{\Lambda}^2(t,F(0,T))}\right) w(t)\mathrm{d}t,$$

and the expression for δ_F is proven.

$\square$

With this theorem the problem of approximately pricing options on the FX-rate in the model (3.1) is solved.

4 Implementation

In previous sections we got familiar with the financial product – Power Reverse Dual Currency swaps, and studied a multi-currency model, that how can it be used for pricing such securities, like PRDCs. In this section, I am going to look through the implementation of Section **3**, the challenge of calibration and the numerical solving of the PDE given for the price of any product whose payoff is a function of the FX-rate and the interest rates of the two currencies.

4.1 Calibration of the parameters

If we want to use a model for valuation, the appearing parameters must be calibrated to real market data, i.e. they need to be chosen to match the prices from the market of related securities.

The volatility structures of the zero coupon bonds in both currencies have the parameters $\sigma_d,\ \sigma_f,\ \varkappa_d,\ \varkappa_f$. They are chosen to match European swaption values in the respective currencies. A swaption (Swap Option) is an option on an interest rate swap, i.e. it reserves the right to purchase a swap at a specific time and interest rate in the future. For detailed information about swaptions and their pricing see [9].
A swaption is an option on a forward interest rate. From the implied volatilities in the two currencies we get two tables, where the rows indicate the option maturity, the columns are for the different swap tenors. For details see [11]. As a first step, we have to calculate the volatilities for every one-year interval, e.g. we have the volatility 1Y2Y (i.e. when the option maturity is 1 year and the swap tenor is 2 years) and 2Y1Y and we want to determine the volatility for the interval [1Y,2Y] from them. Let $\tilde{\sigma}_i$ be the volatility for the ith 'one-year' period.

$$\tilde{\sigma}_{i+1}^2 = (i+1)\sigma_{i+1,i+2}^2 - i\sigma_{i,i+2}^2/(i+1);$$

Although we have the implied volatilities for every one-year period for the forward interest rates, we want to define the volatility of the *logarithm* of the zero coupon bond. The relationship between them is described below, with the notation $f(t,T)$ for the forward interest rate and $P(t,T)$ is for the zero coupon bond. Assume that the forward rate has the dynamics

$$\mathrm{d}f(t,T) = \mu(t,T)\mathrm{d}t + \sigma(t,T)\mathrm{d}W(t).$$

As $P(t,T) = \exp\left\{-\int_t^T f(t,u)\mathrm{d}u\right\}$, we have

$$\mathrm{d}\log P(t,T) = \mathrm{d}\left(-\int_t^T f(t,u)\mathrm{d}u\right) = f(t,t)\mathrm{d}t - \int_t^T \mathrm{d}f(t,u)\mathrm{d}u.$$

The volatility coming form the above is

$$\int_t^T [\sigma(t,u)\mathrm{d}W(u)]\,\mathrm{d}u = \left[\int_t^T \sigma(t,u)\mathrm{d}u\right]\mathrm{d}W(u).$$

The result justifying this interchange of the order of integration is known as the Stochastic Fubini Theorem.

From the above results we have the value of the zero coupon volatilities, and we have the formula for them as well (3.2) with 2-2 parameters for each currency. We have to calibrate the parameters to fit the given values 'the best'. Here 'the best' means to minimize the sum of the square differences between the two values. We have to solve nonlinear least-squares curve fitting problems of the form $\min_x (f_1(x)^2 + f_2(x)^2 + \cdots + f_n(x)^2)$. The algorithm which was used in MATLAB[5] is a subspace trust region method and is based on the interior-reflective Newton method (see [10] for details). The idea of the trust region method is to approximate the function, which we want to minimize, with a simpler function, which reflects the behavior of the original function in a neighborhood, around the point x. This neighborhood is the trust region. The algorithm in each iteration involves the approximate solution of a large linear system using the method of 'preconditioned conjugate gradients' (see e.g. [16]).

The correlation parameters ρ_{df}, ρ_{dS}, ρ_{fS} are also chosen by historical estimation. Assume that we have historical data for equidistant points for all the three

[5] The MATLAB command which is for the purpose of parameter calibration is `lsqnonlin`. As we need upper and lower bounds for the parameters (1 and 0 respectively), the 'Large Scale' method cannot be switched off.

processes $(r_d,\ r_f, S)$. For the FX rate we actually have the dynamics for $\log S(t)$ (3.1). Thus, after we get the logarithm of the time-series data, then we can approximate the drift with $(\log S(n) - \log S(1))/n$, and the volatility with the quadratic variation. For the parameter estimation of the short rates in each currency (3.5) Maximum Likelihood estimation can be used, based on [13]. Assume equivalent dynamics with (3.5) for the short rates, such as

$$\mathrm{d}r_i = \kappa(\mu - r_i)\mathrm{d}t + \sigma \mathrm{d}W.$$

We have a data set $(x_0, x_1, x_2, \dots, x_n)$ at time points $(t_0, t_1, t_2, \dots, t_n)$. Then the ML estimation for the parameters is the following:

$$\tilde{\kappa} = \frac{n\sum_{i=1}^{n} x_{i-1}x_i - n\sum_{i=1}^{n} x_{i-1}^2 - (x_n - x_0)\sum_{i=1}^{n} x_{i-1}}{\left(\left(\sum_{i=1}^{n} x_{i-1}\right)^2 - n\sum_{i=1}^{n} x_{i-1}^2\right)\mathrm{d}t},$$

$$\tilde{\mu} = \frac{x_n - x_0 + \tilde{\kappa}\mathrm{d}t\sum_{i=1}^{n} x_{i-1}}{n\tilde{\kappa}\mathrm{d}t},$$

$$\tilde{\sigma}^2 = \frac{\sum_{i=1}^{n}\left(x_i - x_{i-1} - \tilde{\kappa}(\tilde{\mu} - x_{i-1})\mathrm{d}t\right)^2}{n\mathrm{d}t}.$$

After the parameter estimation, we have equations

$$\begin{aligned} \log S_t &= \log S_{t-1} + drift\Delta t + vol\Delta t N_t^S, \\ ri_t &= ri_{t-1} + drift\Delta t + vol\Delta t N_t^{r_i}, \quad i = d, f, \end{aligned}$$

where 'drift' and 'vol' are the appropriate numbers from the parameter estimation and N_t are normally distributed. We need to calculate the correlation between these normally distributed vectors $\{N^S\}_{t=0}^n,\ \{N^{r_d}\}_{t=0}^n,\ \{N^{r_f}\}_{t=0}^n$.[6]

We have seen what parametrization is used for the $\gamma(t,x)$ local volatility function (3.3), which means in the aspect of calibration, that the time-dependent functions $\nu(t),\ \beta(t)$ is needed to be determined. The function $L(t)$ is

[6] In MATLAB the command `corrcoef` can be used.

chosen to be the forward FX rate, $L(t) = F(0,t),\ t \geq 0$. Assume, that the maturities are given as

$$0 = T_0 < T_1 < \cdots < T_N,$$

and the parameters are to be calibrated to the market prices of options on the FX rate with maturities $\{T_n\}_{n=1}^N$ (see [3] Section 8.). We can choose $\nu(t)$ and $\beta(t)$ to be step-functions i.e. constant between maturities

$$\begin{aligned} \nu(t) &= \sum_{n=1}^N \nu_n \chi_{(T_{n-1},T_n]}(t), \\ \beta(t) &= \sum_{n=1}^N \beta_n \chi_{(T_{n-1},T_n]}(t). \end{aligned}$$

In Theorem 3.5., the result was the approximation of the forward FX rates by a displaced-diffusion process. It is natural, to express the market prices of FX options in the same form. This means, that for each maturity, we have a market volatility σ_n^* and a market skew parameter δ_n^* that if we write the displaced-diffusion model of the form of

$$\mathrm{d}F(t,T) = \sigma_n^*(\delta_n^* F(t,T) + (1-\delta_n^*)F(0,T))\mathrm{d}W_F(t), \quad n = 1,\ldots,N,$$

then we need to chose $\sigma_n^*,\ \delta_n^*$ that the market prices of FX options with expiry T_n across a collection of strikes are being well-matched. This can be done by the volatilities fitted to at-the-money volatilities, and the skews to match the slopes of the market FX volatility smiles for each expiry. After we have the market prices expressed with $\{(\sigma_n^*,\ \delta_n^*)\}_{n=1}^N$, we need to set the model parameters $\{(\nu_n,\ \beta_n)\}_{n=1}^N$ in order to the 'effective' volatility $\sigma_F = \sigma_F(T_n)$ from formula (3.19), and the 'effective' skew $\delta_F = \delta_F(T_n)$ from formula (3.17) computed from the model parameters, match the market-implied values $\{(\sigma_n^*,\ \delta_n^*)\}_{n=1}^N$. In practice this means an algebraic root-search problem and it can be split into N sequential problems, as e.g. $\sigma_F(T_1)$ and $\delta_F(T_1)$ only depend on $\nu_1,\ \beta_1$. So they can be calculated from the two equations

$$\begin{aligned} \sigma_F(T_1) &= \sigma_1^*, \\ \delta_F(T_1) &= \delta_1^*. \end{aligned}$$

$\sigma_F(T_2),\ \delta_F(T_2)$ depend on $\nu_1,\ \nu_2,\ \beta_1,\ \beta_2$, and the values of $\nu_1,\ \beta_1$ can be used from the first step, so the problem is reduced again to a two-dimensional root-search problem. This can be done recursively, until all model parameters $\{(\nu_n,\ \beta_n)\}_{n=1}^N$ are found.

4.2 Numerical solving of the PDE

In the previous section we have already seen the PDE 3.7 which is needed to be solved. Let's see it again, with a new notation $\mathcal{L}$ for the differential operator.

$$\begin{aligned} V_t + \mathcal{L}V \equiv\ & V_t + (\theta_d(t) - \varkappa_d(t) r_d) V_{r_d} + \\ & + (\theta_f(t) - \rho_{fS}\sigma_f(t)\gamma(t, S(t)) - \varkappa_f(t) r_f) V_{r_f} + \\ & + (r_d - r_f) S V_S + \\ & + \tfrac{1}{2}\sigma_d^2(t) V_{r_d r_d} + \tfrac{1}{2}\sigma_f^2(t) V_{r_f r_f} + \tfrac{1}{2}\gamma^2(t, S) S^2 V_{SS} + \\ & + \rho_{df}\sigma_d(t)\sigma_f(t) V_{r_d r_f} + \rho_{dS}\sigma_d(t)\gamma(t, S) S V_{r_d S} + \\ & \quad + \rho_{fS}\sigma_f(t)\gamma(t, S) S V_{r_f S} - r_d V = 0. \end{aligned}$$

Solving this PDE numerically has several challenges: time progress is in the opposite direction than usual; the space is three-dimensional; the coefficients depend on time; all of the cross-derivatives can be found; and the run-time of the program should also be optimized. The main ideas of the solution can be seen in [6].

This equation is solved backward in time, as we know the value of a security at the last payment (we will exactly define it later), so for the sake of simplicity the change of variable $\tau = T_{max} - t$ is used. Then the PDE has the form of $V_\tau = \mathcal{L}V$. Originally, the pricing is defined in an unbounded domain

$$\{(s, r_d, r_f, \tau) | s \geq 0,\ r_d \geq 0,\ r_f \geq 0, \tau \in [0, T_{max}]\}.$$

We want to use Finite Difference (called FD) approximations for the space variables, so a finite-sized domain is used

$$\{(s, r_d, r_f, \tau) \in [0, S] \times [0, R_d] \times [0, R_f] \times [0, T_{max}]\} \equiv \Omega \times T_{max},$$

where $S = 3s(0),\ R_d = 3r_d(0),\ R_f = 3r_f(0)$. Regarding the boundary conditions, Dirichlet-type 'stopped process' boundary conditions were used, i.e. we stop the processes $s(t),\ r_d(t),\ r_f(t)$ when one of them hits the boundary. Then we assume, that they are constant for the life-time of the product, which left after (at least) one process reached the boundary. This means, that we used the discounted payoff for the current values of the state variables.

4.2.1 Implementation of the Crank-Nicolson scheme

The discretization of the PDE is needed to be done first. We have $N_x,\ N_y,\ N_z$ and N_t grid points in the directions of $s,\ r_d,\ r_f,\ \tau$. We denote the uniform grid stepsizes by $h_i = \frac{i_{max}}{N_i+1}$. Let $V^m_{i,j,k}$ be the grid point value of a FD approximation, i.e. $V^m_{i,j,k} \approx V(s_i, r_d j, r_f k, \tau_m) = V(ih_x, jh_y, kh_z, mh_t)$. where $i = 1 \dots N_x,\ j = 1 \dots N_y,\ k = 1 \dots N_z,\ m = 1 \dots N_t + 1$. We used the following FD approximations (the example is for the s variable):

$$\begin{aligned} \frac{\partial V}{\partial s} &\approx \frac{V^m_{i+1,j,k} - V^m_{i-1,j,k}}{2h_x}, \\ \frac{\partial^2 V}{\partial s^2} &\approx \frac{V^m_{i+1,j,k} - 2V^m_{i,j,k} + V^m_{i-1,j,k}}{h_x^2}, \\ \frac{\partial^2 V}{\partial s \partial r_d} &\approx \frac{V^m_{i+1,j+1,k} + V^m_{i-1,j-1,k} - V^m_{i-1,j+1,k} - V^m_{i+1,j-1,k}}{4h_x h_y}. \end{aligned} \tag{4.20}$$

In the differential operator $\mathcal{L}$ every spatial derivative is replaced by its corresponding FD scheme (like (4.20)). The FD discretization of $\mathcal{L}$ at $(s_i, r_d j, r_f k, \tau_m)$ is denoted by $\mathcal{L}V^m_{i,j,k}$. Then, the *Crank-Nicolson scheme* is used to step from time τ_{m-1} to τ_m:

$$\frac{V^m_{i,j,k} - V^{m-1}_{i,j,k}}{\Delta\tau} = \frac{1}{2}\mathcal{L}V^m_{i,j,k} + \frac{1}{2}\mathcal{L}V^{m-1}_{i,j,k},$$

where $i = 1 \dots N_x,\ j = 1 \dots N_y,\ k = 1 \dots N_z$. If $\mathbf{u}^m$ is the vector of the approximated values at time τ_m on the mesh Ω, the Crank-Nicolson method gives us the following equation for $m = 1 \dots N_t$

$$\left(\mathbf{I} - \frac{1}{2}\Delta\tau\mathbf{A}^m\right)\mathbf{u}^m = \left(\mathbf{I} + \frac{1}{2}\Delta\tau\mathbf{A}^{m-1}\right)\mathbf{u}^{m-1} + \frac{1}{2}\Delta\tau\left(\mathbf{g}^m + \mathbf{g}^{m-1}\right), \tag{4.21}$$

where $\mathbf{I}$ is the $N_x N_y N_z \times N_x N_y N_z$ identity matrix and $\mathbf{A}^m$ is the same sized matrix coming from the FD discretization of the differential operator $\mathcal{L}$. The matrix $\mathbf{A}^m$ can be written in an explicit form using tensor products. Let $\mathbf{I}_n$ be the n-dimension identity matrix, $\mathbf{T}_n$ be the tridiagonal matrix with entries $\{1, -2, 1\}$, and $\mathbf{Q}_n$ be the tridiagonal matrix with entries $\{1, 0, -1\}$. Then the coefficient matrix can be written in the following form (with $n = N_x,\ p = N_y,\ q = N_z$):

$$
\begin{aligned}
\mathbf{A}^m(t) \quad = \quad & (r_d - r_f)S\frac{1}{2h_x}(\mathbf{I}_q \otimes \mathbf{I}_p \otimes \mathbf{Q}_n) + \\
& +(\theta_d(t) - \varkappa_d(t)r_d)\frac{1}{2h_y}(\mathbf{I}_q \otimes \mathbf{Q}_p \otimes \mathbf{I}_n) + \\
& +(\theta_f(t) - \rho_{fS}\sigma_f(t)\gamma(t,S(t)) - \varkappa_f(t)r_f)\frac{1}{2h_z}(\mathbf{Q}_q \otimes \mathbf{I}_p \otimes \mathbf{I}_n) + \\
& +\frac{1}{2}\gamma^2(t,S)S^2\frac{1}{h_x^2}(\mathbf{I}_q \otimes \mathbf{I}_p \otimes \mathbf{T}_n) + \\
& +\frac{1}{2}\sigma_d^2(t)\frac{1}{h_y^2}(\mathbf{I}_q \otimes \mathbf{T}_p \otimes \mathbf{I}_n) + \frac{1}{2}\sigma_f^2(t)\frac{1}{h_z^2}(\mathbf{T}_q \otimes \mathbf{I}_p \otimes \mathbf{I}_n) + \\
& +\rho_{dS}\sigma_d(t)\gamma(t,S)S\frac{1}{4h_xh_y}(\mathbf{I}_q \otimes \mathbf{Q}_p \otimes \mathbf{Q}_n) + \\
& +\rho_{fS}\sigma_f(t)\gamma(t,S)S\frac{1}{4h_xh_z}(\mathbf{Q}_q \otimes \mathbf{I}_p \otimes \mathbf{Q}_n) + \\
& +\rho_{df}\sigma_d(t)\sigma_f(t)\frac{1}{4h_yh_z}(\mathbf{Q}_q \otimes \mathbf{Q}_p \otimes \mathbf{I}_n) - r_d\mathbf{I}_{npq},
\end{aligned}
$$

where

$$
\mathbf{T} \quad = \quad \begin{pmatrix}
-2 & 1 & 0 & \dots & & & 0 \\
1 & -2 & 1 & 0 & \dots & & 0 \\
0 & 1 & -2 & 1 & 0 & \dots & 0 \\
\vdots & & \ddots & \ddots & \ddots & \dots & \vdots \\
0 & \dots & & 0 & 1 & -2 & 1
\end{pmatrix}
$$

$$
\mathbf{Q} \quad = \quad \begin{pmatrix}
0 & -1 & 0 & \dots & & & 0 \\
1 & 0 & -1 & 0 & \dots & & 0 \\
0 & 1 & 0 & -1 & 0 & \dots & 0 \\
\vdots & & \ddots & \ddots & \ddots & \dots & \vdots \\
0 & \dots & & 0 & 1 & 0 & -1
\end{pmatrix}.
$$

The method is implicit, i.e. in every time-step we need to solve a linear system to get the actual value for $\mathbf{u}^m$.

The calculations were done in MATLAB 7.1.

For the MATLAB implementation there were some main ideas which were followed in order to simplify the program code. The whole code can be found in the Appendix (Section 5). The first step to start with is that, as it can be seen above, there is a $N_xN_yN_z$-size vector $\mathbf{u}^m$. The solution of the boundary condition was much easier with the collection of the indexes which are effected by the boundary conditions. To have an acceptable run-time, cycles should have been avoided, and vector operations were used, when it was possible. Also the `sparse` command appears often, as it provides good storage conditions for matrices with many zero elements. In order to solve the linear system, the simple '\' command was implemented, and did not calculated the matrix inverse.

4.2.2 Pricing a PRDC swap

Now we need to set the trade specific parameters, i.e. how can the PDE be implemented for a PRDC trade. Suppose that the tenor structure is the following:

$$0 = T_0 < T_1 < \cdots < T_{\beta-1} < T_\beta = T_{max}, \quad \nu_\alpha = \nu(T_{\alpha-1}, T_\alpha) = T_\alpha - T_{\alpha-1}, \quad \alpha = 1, 2, \ldots, \beta-1.$$

That means, there is a payment on every $T_i \quad i = 1, 2, \ldots, \beta - 1$, and the PRDC coupon rate C_α issued at time T_α for the period $[T_\alpha, T_{\alpha+1}]$ is equals to $\nu_\alpha C_\alpha N_d$, as ν_α is the day count fraction using the Actual/365 day count basis, and N_d is the domestic currency principal. We have the coupon definition as it is in the previous section

$$h \max(S(t) - k, 0), \quad \text{where } k = K\frac{jpycpn}{usdcpn}, \quad h = \frac{usdcpn}{K},$$

with the assumption that $floor = 0$; $cap = +\infty$. The payment of the funding leg for the period $[T_{\alpha-1}, T_\alpha]$ is $\nu_\alpha L_d(T_{\alpha-1}, T_\alpha) N_d$, where $L_d(T_{\alpha-1}, T_\alpha) = \dfrac{1 - P_d(T_{\alpha-1}, T_\alpha)}{\nu_\alpha P_d(T_{\alpha-1}, T_\alpha)}$. Note that the floating payments are 'in arrears' in most cases, which means, that the LIBOR rate is observed at time $T_{\alpha-1}$ for the period $[T_{\alpha-1}, T_\alpha]$. Assume that we have annual payments, so $\nu_\alpha \equiv 1$.

For the 'vanilla' PRDCs (often called Bullet PRDCs) the valuation is quite simple, but it is needed to be done before one can move forward to pricing exotic products. Let $V_\alpha^c(t)$ and $V_\alpha^f(t)$ denote the value at time t of all PRDC coupons and floating coupons of the PRDC swap, which are paid on or after $T_{\alpha+1}$. The payoff of the PRDC coupon part at T_α is

$$V_\alpha^c(T_\alpha) + \nu_\alpha C_\alpha N_d.$$

The value of the payoff at $T_{\alpha-1}$ can be obtained by solving the PDE backward from T_α to $T_{\alpha-1}$. The terminal condition at $T_{\beta-1}$ (note that with the variable change described before, this is the condition for $\tau = 1$) is the following

$$V^c_{\beta-1}(T_{\beta-1}) = \nu_{\beta-1} C_{\beta-1} N_d.$$

We can get $V^c_0(T_0)$ by progressing backward to T_0. The floating payment's value at T_0 is

$$1 - P_d(T_0, T_{max}),$$

because, with $B(t)$ denoting the bond price,

$$B(t) E_{\mathbf{P}} \left(\sum_{\alpha=1}^{T_{max}} \frac{\nu_\alpha(L(T_{\alpha-1}, T_\alpha))}{B(T_\alpha)} \middle| F(t) \right) =$$

$$= B(t) \sum_{\alpha=1}^{T_{max}} E_{\mathbf{P}} \left(\frac{1 - P_d(T_{\alpha-1}, T_\alpha)}{P_d(T_{\alpha-1}, T_\alpha)} E_{\mathbf{P}} \left(\frac{1}{B(T_\alpha)} \middle| F(T_{\alpha-1}) \right) \middle| F(t) \right) =$$

$$= B(t) \sum_{\alpha=1}^{T_{max}} E_{\mathbf{P}} \left(\frac{1 - P_d(T_{\alpha-1}, T_\alpha)}{B(T_{\alpha-1})} \middle| F(t) \right) =$$

$$= \sum_{\alpha=1}^{T_{max}} \left(P_d(t, T_{\alpha-1}) - P_d(t, T_\alpha) \right) = p(t, T_0) - p(t, T_{max}),$$

which for $t = T_0$ is $1 - p(T_0, T_{max})$. The value of the PRDC swap is $V^f_0(T_0) - V^c_0(T_0)$. The solution was computed with the following input data:

			period	$\nu(t)$	$\beta(t)$
$P_d(0,T)$	$=$	$\exp(-0.02T)$;	$(0, 0.5]$	9.03%	−200%
$P_f(0,T)$	$=$	$\exp(-0.05T)$;	$(0.5, 1]$	8.87%	−172%
$\sigma_d(t)$	$=$	0.7%;	$(1, 3]$	8.42%	−115%
$\varkappa_d(t)$	$=$	0.00%;	$(3, 5]$	8.99%	−65%
$\sigma_f(t)$	$=$	1.2%;	$(5, 7]$	10.18%	−50%
$\varkappa_f(t)$	$=$	5.0%;	$(7, 10]$	13.31%	−24%
ρ_{df}	$=$	25.0%;	$(10, 15]$	18.18%	10%
ρ_{dS}	$=$	−15.00%;	$(15, 20]$	16.73%	38%
ρ_{fS}	$=$	−15.00%;	$(20, 25]$	13.51%	38%
$S(0)$	$=$	105.00;	$(25, 30]$	13.51%	38%

From $\nu(t)$, $\beta(t)$ and with $L(t) = F(0,t)$ the value of the local volatility function $\gamma(t, S(t))$ can be computed based on (3.3). The parameters $\theta_i(t)$ also can be computed from the above data with (3.6). Note that only these two of the coefficients depend on t in our example. The coupon parameters were chosen to $jpycpn = 4.36\%$; $usdcpn = 6.25\%$.

With this input, and with 6 grid points for each spacial variable, the value of the PRDC swap is -17.45%. This is expressed as a percentage of the notional N_d. The negative value of the PRDC swap determines the price that the investor has to pay for the PRDC coupon payer to enter into the 'vanilla' PRDC swap.

4.2.3 Pricing a Callable PRDC

It is worth mentioning how can all of these be used for an exotic product like a Callable PRDC. (For more details see [6].) The actual implementation is out of the scope of this thesis, but for a future improvement it is an ideal topic. The main idea of valuing a cancellable swap is that terminating the underlying PRDC means the same with continuing the original swap and at the same time entering into the offsetting swap (i.e. the same swap but with the opposite pay-recieve direction). The long position in a Bermudan swaption with the underlying asset being the offsetting swap is called for now the offsetting Bermudan swaption. Let $V_\alpha^e(t)$ be the value at time t of all fund flows in the offsetting swap, i.e. $V_\alpha^e(T_\alpha) = -\left(V_\alpha^f(T_\alpha) - V_\alpha^c(T_\alpha)\right)$. This can be understood as the 'exercise value'. Denote with $V_\alpha^h(t)$ the value at time t of the offsetting Bermudan swaption that has the exercise opportunities on dates $\{T_{\alpha+1}, \ldots, T_{\beta-1}\}$. This is the 'hold value' of the option. Then the payoff of the offsetting Bermudan swaption at T_α is $\max\left(V_\alpha^h(T_\alpha), V_\alpha^e(T_\alpha)\right)$. We can solve the PDE backward in time just as it was the case with the 'vanilla' PRDC, with the terminal condition

$$V_{\beta-1}^h(T_{\beta-1}) = V_{\beta-1}^e(T_{\beta-1}) = 0.$$

Then we can get the value of a Callable PRDC, which is $V_0^h(T_0) + \Big(V_0^f(T_0) - V_0^c(T_0)\Big)$.

5 Appendix

In this appendix the MATLAB codes can be found which were referred to in the previous section.

The basic program `cranknicolson.m` contains the input data, the discretization of the Ω finite spacial domain, the definition of the $\mathbf{T}$, $\mathbf{Q}$ tridiagonal matrices, the collection of the indices needed for the boundary conditions, and the cycle of the time-steps, with calling the `getAcoeffmx.m` and the `getgboundaryvector.m` for calculating the appropriate $\mathbf{A}^m$ coefficient matrix and $\mathbf{g}^m$ boundary condition vector respectively.

`cranknicolson.m`

```
S=315;%FX rate upper bound
Rd=0.06; %JPY interest rate
Rf=0.15; %USD

%constants
sigmad=0.007;
khid=0;
sigmaf=0.012;
khif=0.05;
ro=[0.25 -0.15 -0.15];
spotfx=105;
spotrd=0.02;
spotrf=0.05;
R=spotrf-spotrd;
Tmax=30;%maxtenor

% time dependent parameters
tetad=zeros(1,Tmax);
for i=1:Tmax
    tetad(i)=-R*getFwdFX(i-1,spotfx,R);%khid=0;
```

```
end
tetaf=zeros(1,Tmax);
for i=1:Tmax
    tetaf(i)=-R*getFwdFX(i-1,spotfx,R)+khif*getFwdFX(i-1,spotfx,R)+
        sigmaf^2*(1-exp(-2*khif*(i-1)))/(2*khif);
end

%number of gridpoints
Nx=6;
Ny=6;
Nz=6;

n=Nx*Ny*Nz;

hx=S/(Nx+1);
hy=Rd/(Ny+1);
hz=Rf/(Nz+1);

%s vector
a=(1:Nx)';
b=hx*ones(Ny,1);
c=ones(Nz,1);
s=kron(c,kron(b,a));

%rd vector
d=(1:Ny)';
e=hy*ones(Nx,1);
f=ones(Nz,1);
rd=kron(f,kron(d,e));
```

```
%rf vector
g=(1:Nz)';
h=hz*ones(Nx,1);
i=ones(Ny,1);
rf=kron(g,kron(i,h));

%gamma (Nx x Ny x Nz) x 30 matrix
nu=[0.0903 0.0887 0.0842 0.0842 0.0899 0.0899 0.1018
0.1018 0.133 0.133 0.133 0.1818 0.1818 0.1818 0.1818
0.1818 0.1673 0.1673 0.1673 0.1673 0.1673 0.1351 0.1351
0.1351 0.1351 0.1351 0.1351 0.1351 0.1351 0.1351];
beta=[-2 -1.72 -1.15 -1.15 -0.65 -0.65 -0.5 -0.5 -0.24
-0.24 -0.24 0.1 0.1 0.1 0.1 0.1 0.38 0.38 0.38 0.38
0.38 0.38 0.38 0.38 0.38 0.38 0.38 0.38 0.38 0.38];
gamma=zeros(Nx*Ny*Nz,Tmax);
for i=1:Tmax
    L=getFwdFX(i-1,spotfx,R);
    gamma(:,i)=(nu(i)*(s/L).^(beta(i)-1));
end

%identity matrices
Ix=sparse(eye(Nx));
Iy=sparse(eye(Ny));
Iz=sparse(eye(Nz));

%tridiagonal matrices
Qx=sparse(toeplitz([0,1,zeros(1,Nx-2)],[0,-1,zeros(1,Nx-2)]));
Qy=sparse(toeplitz([0,1,zeros(1,Ny-2)],[0,-1,zeros(1,Ny-2)]));
Qz=sparse(toeplitz([0,1,zeros(1,Nz-2)],[0,-1,zeros(1,Nz-2)]));
```

```
Tx=sparse(toeplitz([-2,1,zeros(1,Nx-2)],[-2,1,zeros(1,Nx-2)]));
Ty=sparse(toeplitz([-2,1,zeros(1,Ny-2)],[-2,1,zeros(1,Ny-2)]));
Tz=sparse(toeplitz([-2,1,zeros(1,Nz-2)],[-2,1,zeros(1,Nz-2)]));

%BOUNDARY CONDITIONS

%collecting indeces
%x right
xperem_jobb_helyek=[];
xpj2d = [Nx:Nx:Nx*Ny];
for i=0:(Nz-1)
xperem_jobb_helyek = [xperem_jobb_helyek,xpj2d+i*Nx*Ny];
end

%x left
xperem_bal_helyek = [];
xpb2d = [1:Nx:(Ny-1)*Nx+1];
for i=0:(Nz-1)
xperem_bal_helyek = [xperem_bal_helyek,xpb2d+i*Nx*Ny];
end

%y back
yperem_hatul_helyek=[];
yph2d = [(Ny-1)*Nx+1:Ny*Nx];
for i=0:(Nz-1)
yperem_hatul_helyek = [yperem_hatul_helyek,yph2d+i*Nx*Ny];
end

%y front
```

```
yperem_elol_helyek=[];
ype2d = [1:Nx];
for i=0:(Nz-1)
yperem_elol_helyek = [yperem_elol_helyek,ype2d+i*Nx*Ny];
end

%z down
zperem_lent_helyek=[1:Nx*Ny];

%z up
zperem_fent_helyek=[(Nz-1)*Nx*Ny+1:Nz*Nx*Ny];

%product specific inputs
%low leverage
%cd=0.0225;
%cf=0.045;

%medium leverage
cd=0.0436;
cf=0.0625;

%high leverage
%cd=0.081;
%cf=0.09;

t=1;% Tmax-t

C29=getCoupon(t,Tmax,spotfx,R,cd,cf,s);

u=C29;
```

```
for t=2:Tmax
    T_alfa2=Tmax-t;%index of V^k
    A_t=getAcoeffmx(T_alfa2,n,s,rd,rf,sigmad,khid,sigmaf,khif,ro,
        tetad,tetaf,gamma,Ix,Iy,Iz,Qx,Qy,Qz,Tx,Ty,Tz);
    baloldal=eye(n)-0.5*A_t;

    T_alfa1=Tmax-t+1;
    A_t_1=getAcoeffmx(T_alfa1,n,s,rd,rf,sigmad,khid,sigmaf,khif,ro,
        tetad,tetaf,gamma,Ix,Iy,Iz,Qx,Qy,Qz,Tx,Ty,Tz);
    g_t=getgboundaryvector(t,Tmax,Nx,Ny,Nz,n,spotfx,spotrd,R,cd,cf,S,
        s,rd,rf,hx,hy,hz,sigmad,khid,sigmaf,khif,ro,tetad,tetaf,gamma,
        xperem_jobb_helyek,xperem_bal_helyek,
        yperem_elol_helyek,yperem_hatul_helyek,
        zperem_lent_helyek,zperem_fent_helyek);
    g_t_1=getgboundaryvector(t,Tmax,Nx,Ny,Nz,n,spotfx,spotrd,R,cd,cf,S,
        s,rd,rf,hx,hy,hz,sigmad,khid,sigmaf,khif,ro,tetad,tetaf,gamma,
        xperem_jobb_helyek,xperem_bal_helyek,
        yperem_elol_helyek,yperem_hatul_helyek,
        zperem_lent_helyek,zperem_fent_helyek);
    jobboldal=(eye(n)+0.5*A_t_1)*u + 0.5*(g_t+g_t_1);

    u=baloldal\jobboldal;

end

price=1-exp(-spotrd*Tmax)-u;}
```

getAcoeffmx.m

```
function A = getAehmx(t,n,s,rd,rf,sigmad,khid,sigmaf,khif,ro,
        tetad,tetaf,gamma,Ix,Iy,Iz,Qx,Qy,Qz,Tx,Ty,Tz)
```

```
elso_matrix=sparse(diag((rd-rf).*s)*kron(Iz,kron(Iy,Qx)));
masodik_matrix=getMx2(t,n,tetad,khid,rd,Qy,Ix,Iz);
harmadik_matrix=getMx3(t,n,tetaf,khif,rf,ro,sigmaf,gamma,Qz,Iy,Ix);
negyedik_matrix=getMx4(t,gamma,s,Tx,Iy,Iz);
otodik_matrix=sparse(0.5*sigmad^2*kron(Iz,kron(Ty,Ix)));
hatodik_matrix=sparse(0.5*sigmaf^2*kron(Tz,kron(Iy,Ix)));
hetedik_matrix=getMx7(t,ro,sigmad,gamma,s,Qx,Qy,Iz);
nyolcadik_matrix=getMx8(t,ro,sigmaf,gamma,s,Qx,Iy,Qz);
kilencedik_matrix=sparse(ro(1)*sigmad*sigmaf*kron(Qz,kron(Qy,Ix)));
tizedik_matrix=diag(rd);

A=sparse(elso_matrix+masodik_matrix+harmadik_matrix+
          negyedik_matrix+otodik_matrix+hatodik_matrix+hetedik_matrix+
          nyolcadik_matrix+kilencedik_matrix-tizedik_matrix);
end

function mx = getMx2(t,n,tetad,khid,rd,Qy,Ix,Iz);

mx=sparse(diag(tetad(t+1)*ones(n,1)-khid*rd)*kron(Iz,kron(Qy,Ix)));

end

function mx = getMx3(t,n,tetaf,khif,rf,ro,sigmaf,gamma,Qz,Iy,Ix)

mx = sparse(diag(tetaf(t+1)*ones(n,1)-khif*rf-
          ro(3)*sigmaf*gamma(:,t+1))*kron(Qz,kron(Iy,Ix)));

end
```

```
function mx = getMx4(t,gamma,s,Tx,Iy,Iz)

mx = sparse(0.5*diag(gamma(:,t+1).^2.*s.^2)*kron(Iz,kron(Iy,Tx)));

end

function mx = getMx7(t,ro,sigmad,gamma,s,Qx,Qy,Iz)

mx = sparse(diag(ro(2)*sigmad*gamma(:,t+1).*s)*kron(Iz,kron(Qy,Qx)));

end

function mx = getMx8(t,ro,sigmaf,gamma,s,Qx,Iy,Qz)

mx = sparse(diag(ro(3)*sigmaf*gamma(:,t+1).*s)*kron(Qz,kron(Iy,Qx)));

end
```

getFwdFX.m

```
function fwdfx = getFwdFX(T,spotfx,R)
 if (T==0)
     fwdfx=spotfx;
 else
     fwdfx = spotfx*exp(-R*T);
 end
end
```

getgboundaryvector.m

```
function gout = getgperemvektor(t,Tmax,Nx,Ny,Nz,n,spotfx,spotrd,R,
                             cd,cf,S,s,rd,rf,hx,hy,hz,
                             sigmad,khid,sigmaf,khif,ro,tetad,tetaf,gamma,
```

```
                              xperem_jobb_helyek,xperem_bal_helyek,
                              yperem_elol_helyek,yperem_hatul_helyek,
                              zperem_lent_helyek,zperem_fent_helyek)

T_alfa=Tmax-t;
discount=exp(-spotrd*T_alfa);
%%%set the boundary conditions
%depending on t
%xbound_right
payoffj=0;
for i=1:t
    payoffj=payoffj+getCoupon(i,Tmax,spotfx,R,cd,cf,S);
end
xperem_jobb=zeros(length(xperem_jobb_helyek),1)+discount*payoffj;

%xbound_left
xperem_bal=zeros(length(xperem_bal_helyek),1);

%ybound_front
s_elol=s(yperem_elol_helyek);
payoffe=zeros(length(yperem_elol_helyek),1);
for j=1:t
    payoffe=payoffe+getCoupon(j,Tmax,spotfx,R,cd,cf,s_elol);
end
yperem_elol=discount*payoffe;

%ybound_back
s_hatul=s(yperem_hatul_helyek);
payoffh=zeros(length(yperem_hatul_helyek),1);
for k=1:t
```

```
    payoffh=payoffh+getCoupon(k,Tmax,spotfx,R,cd,cf,s_hatul);
end
yperem_hatul=discount*payoffh;

%zbound_down
s_lent=s(zperem_lent_helyek);
payoffl=zeros(length(zperem_lent_helyek),1);
for l=1:t
    payoffl=payoffl+getCoupon(l,Tmax,spotfx,R,cd,cf,s_lent);
end
zperem_lent=discount*payoffl;

%zbound_up
s_fent=s(zperem_fent_helyek);
payofff=zeros(length(zperem_fent_helyek),1);
for m=1:t
    payofff=payofff+getCoupon(m,Tmax,spotfx,R,cd,cf,s_fent);
end
zperem_fent=discount*payofff;

%%set the boundary conditions
%%on the sides of the cube for the cross-derivatives shift
xperem_jobb_lap=zeros(Nz+2,Ny+2)+discount*payoffj;
xperem_bal_lap=zeros(Nz+2,Ny+2);

yperem_elol_lap=zeros(Nz+2,Nx+2);
Ke=reshape(yperem_elol,Nx,Nz)';
perm5=[Nz:-1:1];
yperem_elol_lap(2:Nz+1,2:Nx+1)=Ke(perm5,:);
yperem_elol_lap(:,Nx+2)=yperem_elol_lap(:,Nx+2)+discount*payoffj;
```

```
yperem_elol_lap(1,:)=yperem_elol_lap(2,:);
yperem_elol_lap(Nz+2,:)=yperem_elol_lap(2,:);

yperem_hatul_lap=yperem_elol_lap;

zperem_lent_lap=zeros(Ny+2,Nx+2);
Kl=reshape(zperem_lent,Nx,Ny)';
perm14=[Ny:-1:1];
zperem_lent_lap(2:Ny+1,2:Nx+1)=Kl(perm14,:);
zperem_lent_lap(:,Nx+2)=zperem_lent_lap(:,Nx+2)+discount*payoffj;
zperem_lent_lap(1,:)=zperem_lent_lap(2,:);
zperem_lent_lap(Ny+2,:)=zperem_lent_lap(2,:);

zperem_fent_lap=zperem_lent_lap;

%%filling vector g
g=zeros(n,1);

%x first order
g(xperem_jobb_helyek) = g(xperem_jobb_helyek)+
          (rd(xperem_jobb_helyek)-rf(xperem_jobb_helyek)).*
          s(xperem_jobb_helyek).*xperem_jobb/2*hx;
g(xperem_bal_helyek) = g(xperem_bal_helyek)-(rd(xperem_bal_helyek)-
          rf(xperem_bal_helyek)).*s(xperem_bal_helyek).*xperem_bal/2*hx;

%y first order
g(yperem_elol_helyek) = g(yperem_elol_helyek)-
          (tetad(T_alfa+1)-khid*rd(yperem_elol_helyek)).*yperem_elol/2*hy;
g(yperem_hatul_helyek) = g(yperem_hatul_helyek)+
          (tetad(T_alfa+1)-khid*rd(yperem_hatul_helyek)).*yperem_hatul/2*hy;
```

```
%z first order
gammat=gamma(:,T_alfa+1);
g(zperem_lent_helyek) = g(zperem_lent_helyek)-
          (tetaf(T_alfa+1)-khif*rf(zperem_lent_helyek)-
          ro(3)*sigmaf*gammat(zperem_lent_helyek)).*zperem_lent/2*hz;
g(zperem_fent_helyek) = g(zperem_fent_helyek)+
          (tetaf(T_alfa+1)-khif*rf(zperem_fent_helyek)-
          ro(3)*sigmaf*gammat(zperem_fent_helyek)).*zperem_fent/2*hz;

%x second order
gammat=gamma(:,T_alfa+1);
g(xperem_jobb_helyek) = g(xperem_jobb_helyek)+
          (0.5*(gammat(xperem_jobb_helyek).^2).*
          s(xperem_jobb_helyek).^2).*xperem_jobb/hx^2;
g(xperem_bal_helyek) = g(xperem_bal_helyek)+
          (0.5*(gammat(xperem_bal_helyek).^2).*
          s(xperem_bal_helyek).^2).*xperem_bal/hx^2;

%y second order
g(yperem_elol_helyek) = g(yperem_elol_helyek)+
          0.5*sigmad^2*yperem_elol/hy^2;
g(yperem_hatul_helyek) = g(yperem_hatul_helyek)+
          0.5*sigmad^2*yperem_hatul/hy^2;

%z second order
g(zperem_lent_helyek) = g(zperem_lent_helyek)+
          0.5*sigmaf^2*zperem_lent/hz^2;
g(zperem_fent_helyek) = g(zperem_fent_helyek)+
          0.5*sigmaf^2*zperem_fent/hz^2;
```

```
%%%%cross derivatives
%%xy
%%%shifting vectors
%xbound
shift1=xperem_jobb_lap(2:Nz+1,1:Ny);
perm1=[Nz:-1:1];
A1=shift1(perm1,:)';
xperem_jobb_hatra=A1(:);

shift2=xperem_bal_lap(2:Nz+1,1:Ny);
perm2=[Nz:-1:1];
A2=shift2(perm2,:)';
xperem_bal_hatra=A2(:);

shift3=xperem_jobb_lap(2:Nz+1,3:Ny+2);
perm3=[Nz:-1:1];
A3=shift3(perm3,:)';
xperem_jobb_elore=A3(:);

shift4=xperem_bal_lap(2:Nz+1,3:Ny+2);
perm4=[Nz:-1:1];
A4=shift4(perm4,:)';
xperem_bal_elore=A4(:);

%ybound
shift5=yperem_elol_lap(2:Nz+1,1:Nx);
perm5=[Nz:-1:1];
A5=shift5(perm5,:)';
yperem_elol_jobbra=A5(:);
```

```
shift6=yperem_hatul_lap(2:Nz+1,1:Nx);
perm6=[Nz:-1:1];
A6=shift6(perm6,:)';
yperem_hatul_jobbra=A6(:);

shift7=yperem_elol_lap(2:Nz+1,3:Nx+2);
perm7=[Nz:-1:1];
A7=shift7(perm7,:)';
yperem_elol_balra=A7(:);

shift8=yperem_hatul_lap(2:Nz+1,3:Nx+2);
perm8=[Nz:-1:1];
A8=shift8(perm8,:)';
yperem_hatul_balra=A8(:);

%filling g for xy cross derivative
gammat=gamma(:,T_alfa+1);
g(xperem_jobb_helyek)=g(xperem_jobb_helyek)+
          ro(2)*sigmad*gammat(xperem_jobb_helyek).*
          s(xperem_jobb_helyek).*
          ((-xperem_jobb_hatra+xperem_jobb_elore)/4*hx*hy);

g(yperem_hatul_helyek)=g(yperem_hatul_helyek)+
          ro(2)*sigmad*gammat(yperem_hatul_helyek).*
          s(yperem_hatul_helyek).*
          ((yperem_hatul_balra-yperem_hatul_jobbra)/4*hx*hy);

g(xperem_bal_helyek)=g(xperem_bal_helyek)+
          ro(2)*sigmad*gammat(xperem_bal_helyek).*
```

```
        s(xperem_bal_helyek).*
        ((-xperem_bal_elore+xperem_bal_hatra)/4*hx*hy);

g(yperem_elol_helyek)=g(yperem_elol_helyek)+
        ro(2)*sigmad*gammat(yperem_elol_helyek).*
        s(yperem_elol_helyek).*
        ((yperem_elol_jobbra-yperem_elol_balra)/4*hx*hy);

%%xz cross derivative
%%%shifting vectors
%xbound
shift9=xperem_jobb_lap(3:Nz+2,2:Ny+1);
perm9=[Nz:-1:1];
A9=shift9(perm9,:)';
xperem_jobb_fel=A9(:);

shift10=xperem_bal_lap(3:Nz+2,2:Ny+1);
perm10=[Nz:-1:1];
A10=shift10(perm10,:)';
xperem_bal_fel=A10(:);

shift11=xperem_jobb_lap(1:Nz,2:Ny+1);
perm11=[Nz:-1:1];
A11=shift11(perm11,:)';
xperem_jobb_le=A11(:);

shift12=xperem_bal_lap(1:Nz,2:Ny+1);
perm12=[Nz:-1:1];
A12=shift12(perm12,:)';
xperem_bal_le=A12(:);
```

```
%zbound
shift13=zperem_fent_lap(2:Ny+1,1:Nx);
perm13=[Ny:-1:1];
A13=shift13(perm13,:)';
zperem_fent_jobbra=A13(:);

shift14=zperem_lent_lap(2:Ny+1,1:Nx);
perm14=[Ny:-1:1];
A14=shift14(perm14,:)';
zperem_lent_jobbra=A14(:);

shift15=zperem_fent_lap(2:Ny+1,3:Nx+2);
perm15=[Ny:-1:1];
A15=shift15(perm15,:)';
zperem_fent_balra=A15(:);

shift16=zperem_lent_lap(2:Ny+1,3:Nx+2);
perm16=[Ny:-1:1];
A16=shift16(perm16,:)';
zperem_lent_balra=A16(:);

%filling g for xz cross derivative
gammat=gamma(:,T_alfa+1);
g(xperem_jobb_helyek)=g(xperem_jobb_helyek)+
          ro(3)*sigmaf*gammat(xperem_jobb_helyek).*
          s(xperem_jobb_helyek).*
          ((-xperem_jobb_fel+xperem_jobb_le)/4*hx*hz);

g(zperem_fent_helyek)=g(zperem_fent_helyek)+
```

```
            ro(3)*sigmaf*gammat(zperem_fent_helyek).*
            s(zperem_fent_helyek).*
            ((zperem_fent_balra-zperem_fent_jobbra)/4*hx*hz);

g(xperem_bal_helyek)=g(xperem_bal_helyek)+
            ro(3)*sigmaf*gammat(xperem_bal_helyek).*
            s(xperem_bal_helyek).*
            ((-xperem_bal_le+xperem_bal_fel)/4*hx*hz);

g(zperem_lent_helyek)=g(zperem_lent_helyek)+
            ro(3)*sigmaf*gammat(zperem_lent_helyek).*
            s(zperem_lent_helyek).*
            ((zperem_lent_jobbra-zperem_lent_balra)/4*hx*hz);

%%yz cross derivative
%%%shifting vectors
%ybound
shift17=yperem_elol_lap(3:Nz+2,2:Nx+1);
perm17=[Nz:-1:1];
A17=shift17(perm17,:)';
yperem_elol_fel=A17(:);

shift18=yperem_hatul_lap(3:Nz+2,2:Nx+1);
perm18=[Nz:-1:1];
A18=shift18(perm18,:)';
yperem_hatul_fel=A18(:);

shift19=yperem_elol_lap(1:Nz,2:Nx+1);
perm19=[Nz:-1:1];
A19=shift19(perm19,:)';
```

```
yperem_elol_le=A19(:);

shift20=yperem_hatul_lap(1:Nz,2:Nx+1);
perm20=[Nz:-1:1];
A20=shift20(perm20,:)';
yperem_hatul_le=A20(:);

%zbound
shift21=zperem_fent_lap(3:Ny+2,2:Nx+1);
perm21=[Ny:-1:1];
A21=shift21(perm21,:)';
zperem_fent_hatra=A21(:);

shift22=zperem_lent_lap(3:Ny+2,2:Nx+1);
perm22=[Ny:-1:1];
A22=shift22(perm22,:)';
zperem_lent_hatra=A22(:);

shift23=zperem_fent_lap(1:Ny,2:Nx+1);
perm23=[Ny:-1:1];
A23=shift23(perm23,:)';
zperem_fent_elore=A23(:);

shift24=zperem_lent_lap(1:Ny,2:Nx+1);
perm24=[Ny:-1:1];
A24=shift24(perm24,:)';
zperem_lent_elore=A24(:);

%filling g for yz cross derivative
g(yperem_hatul_helyek)=g(yperem_hatul_helyek)+
```

```
            ro(1)*sigmad*sigmaf*
            ((-yperem_hatul_fel+yperem_hatul_le)/4*hy*hz);

g(zperem_fent_helyek)=g(zperem_fent_helyek)+
            ro(1)*sigmad*sigmaf*
            ((zperem_fent_elore-zperem_fent_hatra)/4*hy*hz);

g(yperem_elol_helyek)=g(yperem_elol_helyek)+
            ro(1)*sigmad*sigmaf*
            ((-yperem_elol_le+yperem_elol_fel)/4*hy*hz);

g(zperem_lent_helyek)=g(zperem_lent_helyek)+
            ro(1)*sigmad*sigmaf*
            ((zperem_lent_hatra-zperem_lent_elore)/4*hy*hz);

gout=g;
end
```

References

[1] Chia Chiang Tan: *Demistifying exotic products*, 2010.

[2] Jason Sippel; Shoichi Ohkoshi: *All power to PRDC notes*, Risk, October 2002.

[3] Vladimir V. Piterbarg: *A multi-currency model with FX volatility skew*, 2005.

[4] R. Gibson; F.-S. Lhabitant; D. Talay: *Modeling the term structure of interest rates: a review of the literature*, 2001.

[5] Mag. Martin Vonwald: *Advanced topics in continuous time finance*, 2001.

[6] D. M. Dang; C. C. Christara; K.R. Jackson; A. Lakhany: *A PDE pricing framework for cross-currency interest rate derivatives*, 2010.

[7] B. Dupire: *Pricing with a smile*, 1994.

[8] Vladimir V. Piterbarg: *Time to smile*, 2005.

[9] D. Akume, B. Luderer, G.-W. Weber: *Pricing and hedging of swaptions*, 2003.

[10] T. F. Coleman, Y. Li: *An Interior, Trust Region Approach for Nonlinear Minimization Subject to Bounds*, 1996.

[11] D. Brigo, F. Mercurio: *Interest-Rate Models - Theory and Practice*, 2001.

[12] L. A. Grzelak, Cornelis W. Oosterlee: *On Cross-Currency Models with Stochastic Volatility and Correlated Interest rates* , 2011.

[13] G. Molnár - Sáska: *Maximum Likelihood estimation of mean reversion parameters of GBM*

[14] R. Beneder, M. Elkenbracht-Huizing: *Foreign Exchange options and the Volatility Smile*, 2003.

[15] Vladimir V. Piterbarg: *Stochastic Volatility Model with Time-dependent Skew*, 2005.

[16] E. Caraga: *Preconditioned Conjugate Gradient Algorithm*, BSc Thesis, 2008.