# Valuation of Contingent Convertibles with Derivatives

Nicolay Schmitt

Frankfurt School of Finance and Management

A thesis submitted for the degree of

*Master of Science in Finance*

August 2016

# Abstract

Financial crises have led to higher regulatory standards on the capital adequacy of banks. Banks are required to hold more capital with loss absorbance capacities on their balance sheet. In conjunction with this development, contingent convertible bonds (CoCos) have become an attractive instrument for banks to seek new capital. The defining characteristic of CoCos is the automatic conversion into common equity when a predetermined trigger is met. Loss absorbing capital is created, which instantly improves the capital structure of the distressed bank. However, the pricing of these hybrid instruments remains opaque. In this context, the thesis scrutinizes the valuation of CoCos. Three dominant approaches are examined: the structural approach in accordance to Pennacchi (2010), the credit derivatives approach and the equity derivatives approach both pursuant to De Spiegeleer and Schoutens (2011). The application covers sensitivity analysis to further understand the dynamics of the different methodologies. Based on a case study the viability of those approaches is evaluated. Software is provided for further replication.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction and Motivation

## 1.1 Introduction

| Face Value in USD bn | Issuer |
|---|---|
| 3.75 | HSBC (GB) |
| 3.63 | UBS (CH) |
| 3.15 | Royal Bank of Scotland (GB) |
| 3.00 | Barclays (GB) |
| 2.70 | UBS (CH) |
| 2.50 | Credit Suisse (CH) |
| 2.50 | UBS (CH) |
| 2.45 | HSBC (GB) |
| 2.25 | ING (NL) |
| 2.06 | Banco Santander (ESP) |

Table 1.1: Largest CoCo issues in Europe from 2010 to 2016 (Dietegen, 2016)

## 1.2 Literature Overview

| Structural Approach | Equity Derivative Approach | Credit Derivative Approach |
|---|---|---|
| Pennacchi (2010) | De Spiegeleer and Schoutens (2011) | De Spiegeleer and Schoutens (2011) |
| Glasserman and Nouri (2012) | Henriques and Doctor (2011) | |
| Madan and Schoutens (2011) | | |
| Albul et al. (2010) | | |
| Sundaresan and Wang (2015) | | |
| Hilscher and Raviv (2014) | | |
| Buergi (2013) | | |

Table 1.2: Literature overview of valuation approaches for CoCos (Erismann, 2015)

## 1.3 Motivation

## 1.4 Methodology

# Chapter 2

# Structure of CoCos

## 2.1 Description of CoCos

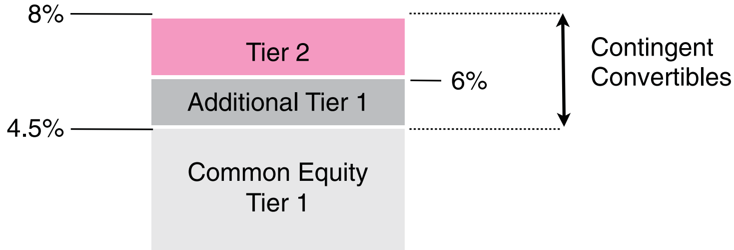


Figure 2.1: CoCos under Basel III (De Spiegeleer et al., 2014)

## 2.2 Payoff and Risk Profile

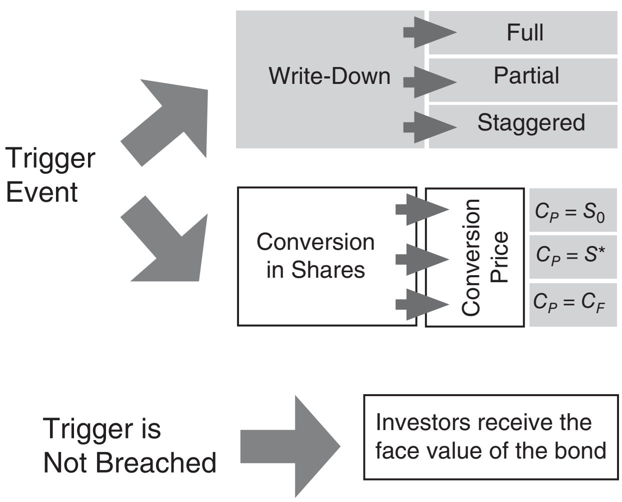


Figure 2.2: Anatomy of CoCos (De Spiegeleer et al., 2014)

## 2.3 Conversion Trigger

### 2.3.1 Market Trigger

### 2.3.2 Accounting Trigger

### 2.3.3 Regulatory Trigger

### 2.3.4 Multivariate Trigger

## 2.4 Conversion Details

### 2.4.1 Conversion Fraction

- conversion fraction $\alpha$
- face value $N$
- conversion amount $N \times \alpha$
- amount remaining in case of partial equity conversion $N \times (1 - \alpha)$

### 2.4.2 Conversion Price and Ratio

- conversion rate $C_r$
- conversion price $C_p$
- recovery rate $R_{CoCo}$
- stock price at trigger event $S_T^*$
- loss attributable to CoCo holders $L_{CoCo}$

$$C_p = \frac{\alpha N}{C_r} \tag{2.1}$$

$$C_r = \frac{\alpha N}{C_p} \tag{2.2}$$

$$R_{CoCo} = \frac{S_T^*}{C_p} \tag{2.3}$$

$$L_{CoCo} = N - (1 - R_{CoCo}) = N \left(1 - \frac{S_T^*}{C_p}\right) \tag{2.4}$$

$$\tag{2.5}$$

$$P_T = \begin{cases} (1-\alpha)N + \frac{\alpha N}{C_p} S_T^* & \text{if converted} \\ N & \text{if not converted} \end{cases} \tag{2.6}$$

# Chapter 3

# Theory of Pricing

## 3.1 Credit Derivative Approach

In financial markets the reduced-form approach is widely used in order to price credit risk. It was originally introduced by Jarrow and Turnbull (1995) respectively Duffie and Singleton (1999). The credit derivative approach applies the reduced-form approach to CoCos. In this context, the derivation of a pricing formula for CoCos follows mainly De Spiegeleer and Schoutens (2011).

### 3.1.1 Reduced-Form Approach and Credit Triangle

The reduced-form approach is an elegant way of bridging the gap between the prediction of default and the pricing of default risk of a straight bond. In the following, we investigate the link between estimated default intensities and credit spreads under the reduced-form approach. (Lando, 2009)

Let $\tau$ denote the random time of default of some company. It is assumed that the distribution of $\tau$ has a continuous density function $f$, so that the distribution function $F$ and the curve of survival probabilities $q$ are related as follows:

$$P(\tau \leq t) = F(t) = 1 - q(t) = \int_0^t f(s)ds, \text{ with } t \geq 0 \tag{3.1}$$

The hazard rate respectively the default intensity $\lambda$ is defined as follows:

$$\lambda(t) = \lim_{\Delta \downarrow 0} \frac{1}{\Delta} P(\tau \leq t + \Delta | \tau > t) = \frac{F'(t)}{q(t)} = \frac{F'(t)}{1 - F(t)} = -\frac{d}{dt} \log q(t) \tag{3.2}$$

Intuitively, the hazard rate is the default rate per year as of today. Using 3.2 we can derive a formula for the survival probability:

$$q(t) = \exp\left(-\int_0^t \lambda(s)ds\right) \tag{3.3}$$

For our application of the reduced-form approach we assume that the hazard rate $\lambda(t)$ is a deterministic function of time. In reality $\lambda(t)$ is not deterministic but itself stochastic. That fits in with the fact that credit spreads are not static but stochastically varying over time. (Schmidt, 2015) However, we further consider the hazard rate to be constant in order to simplify the problem. Hence, a constant hazard rate $\lambda(t) = \lambda$ implies an exponential distribution of the default time:

$$F(t) = 1 - q(t) = 1 - \exp(-\lambda t) \tag{3.4}$$

Under the idealized assumption of a flat zero interest rate curve, a flat spread curve and continuous spread payments, the default intensity $\lambda$ can be calculated directly from the credit spread $s$ and the recovery rate $R$ by the rule of thumb formula (Schmidt, 2015), which is also known as credit triangle:

$$\lambda = \frac{s}{1-R} \tag{3.5}$$

Finally, this relationship makes it possible to determine the default probability $F$ from the credit spread $s$ and vice versa.

### 3.1.2 Adaption to CoCos

In accordance with the aforementioned reduced-form approach, De Spiegeleer and Schoutens (2011) assume that the probability $F^*$, which measures the likelihood that a CoCo triggers within the next $T - t$ years, follows similar mechanics as the default probability of a straight bond does. Under the credit derivative approach the probability $F^*$ can be expressed as follows:

$$F^* = 1 - \exp\left[-\lambda_{Trigger}(T - t)\right] \tag{3.6}$$

Additionally, the credit derivative approach models $F^*$ with the first exit time equation used in barrier option pricing under a Black-Scholes setting. (Su and Rieger, 2009) Hence, the probability $F^*$ that the trigger level $S^*$ is touched within the next $T - t$ years is given by the following equation with the continuous dividend yield $q$, the continuous interest rate $r$, the drift $\mu$, the volatility $\sigma$ and the current share price $S$ of the issuing company:

$$F^* = \Phi\left(\frac{\log\left(\frac{S^*}{S}\right) - \mu(T-t)}{\sigma\sqrt{(T-t)}}\right) + \left(\frac{S^*}{S}\right)^{\frac{2\mu}{\sigma^2}} \Phi\left(\frac{\log\left(\frac{S^*}{S}\right) + \mu(T-t)}{\sigma\sqrt{(T-t)}}\right) \tag{3.7}$$

In this regard, a CoCo's credit spread $s_{CoCo}$ can be approximated by the credit triangle, where $R_{CoCo}$ denotes the recovery rate of a CoCo and $L_{CoCo}$ is the loss rate:

$$s_{CoCo} = (1 - R_{CoCo})\, \lambda_{Trigger} = L_{CoCo} \lambda_{Trigger} \tag{3.8}$$

In the trigger event, the face value $N$ converts into $C_r$ shares worth $S^*$. The loss of a long position in a CoCo is therefore determined by the conversion price $C_p$:

$$Loss_{CoCo} = N - C_r S^* = N\,(1 - R_{CoCo}) = N \left(1 - \frac{S^*}{C_p}\right) \tag{3.9}$$

By combining 3.6, 3.8 and 3.9 we see that the credit spread $s_{CoCo}$ of a CoCo with maturity $T$ at time $t$ is driven by its major design elements, the trigger level $S^*$ and the conversion price $C_p$:

$$s_{CoCo_t} = -\frac{\log(1 - F^*)}{(T - t)} \left(1 - \frac{S^*}{C_p}\right) \tag{3.10}$$

Subsequently, a pricing formula for CoCos under the credit derivative approach can be derived. The present value $V^{cd}$ at time t is given by:

$$V_t^{cd} = \sum_{i=1}^{T} c_i \exp\left[-(r + s_{CoCo_t})(t_i - t)\right] + N \exp\left[-(r + s_{CoCo_t})(T - t)\right] \tag{3.11}$$

In summary, the credit derivative approach provides us with a concise method to price CoCos. However, one has to bear in mind its largest shortcoming. Losses from cancelled coupons of triggered CoCos are not taken into account in the valuation. Hence, the credit derivative approach naturally overestimates the price of CoCos, but it equips investors with a simple rule of thumb formula.

### 3.1.3 Parameter Classification and Adjustment

| | Description | Usage | Source |
|---|---|---|---|
| $T$ | CoCo maturity | Static input | Term sheet |
| $N$ | CoCo nominal | Static input | Term sheet |
| $c$ | CoCo coupon rate | Static input | Term sheet |
| $S_0$ | Initial share price of the issuer | Dynamic input | Market data |
| $S^*$ | Trigger share price | Static input | Term sheet |
| $C_p$ | CoCo nominal conversion price | Static input | Term sheet |
| $r$ | Risk-free interest rate | Static input | Market data |
| $q$ | Dividend yield | Static input | Market data |
| $\sigma$ | Implied volatility | Static input | Market data |

Table 3.1: Parameter classification of the credit derivative approach (Wilkens and Bethke, 2014)

### 3.1.4 Model Application

| | Value | Comment |
|---|---|---|
| $T$ | 10yrs | Maturity |
| $N$ | 100 | Nominal |
| $c$ | 6.00 | Annual coupon rate |
| $S_0$ | 100 | Initial share price of the bank |
| $S^*$ | 35 | Trigger share price |
| $C_p$ | 65 | Nominal conversion price |
| $r$ | 1.00% | Risk-free interest rate |
| $q$ | 2.00% | Dividend yield |
| $\sigma$ | 30.00% | Implied volatility |

Table 3.2: Parameter specification of credit derivative approach application (Alvemar and Ericson, 2012)

## 3.2 Equity Derivative Approach

In order to assess the value of a CoCo, investors can use a method which depends on equity derivatives. (De Spiegeleer and Schoutens, 2011; De Spiegeleer et al., 2014) The so-called equity derivative approach attempts to compensate for the main drawback of the credit derivative approach, since it takes into account that coupon payments might be cancelled if the trigger of a CoCo was touched.

Subsequently, the valuation can be divided into two steps: In the first step, the value of a CoCo is determined without coupon payments. Such a CoCo is called Zero-Coupon CoCo. In the second step, one has to incorporate the coupon payments in the pricing formula while keeping in mind that they might be knocked out. The closed-form solution is derived under a Black-Scholes setting.

### 3.2.1 Step One - Zero-Coupon CoCo

To determine the present value of a Zero-Coupon CoCo $V^{zcoco}$ at maturity $T$ we can use equation 2.6. The underlying assumption of the equity derivative approach is that the triggering of a CoCo respectively of a Zero-Coupon CoCo is equivalent to the share price falling below the level $S^*$.

$$
\begin{aligned}
V_T^{zcoco} &= \begin{cases} N & \text{if not triggered} \\ (1-\alpha)N + \frac{\alpha N}{C_p} S^* & \text{if triggered} \end{cases} \\
&= N \mathbb{1}_{\{\tau > T\}} + \left[ (1-\alpha)\, N + \frac{\alpha N}{C_p} S^* \right] \mathbb{1}_{\{\tau \leq T\}} \\
&= N + \left[ \frac{\alpha N}{C_p} S^* - \alpha N \right] \mathbb{1}_{\{\tau \leq T\}} \\
&= N + \left[ C_r S^* - \alpha N \right] \mathbb{1}_{\{\tau \leq T\}} \\
&= N + C_r \left[ S^* - \frac{\alpha N}{C_r} \right] \mathbb{1}_{\{\tau \leq T\}} \\
&= N + C_r \left[ S^* - C_p \right] \mathbb{1}_{\{\tau \leq T\}} \qquad (3.12)
\end{aligned}
$$

It may be inferred that the financial payoff of equation 3.12 consists of two components (Erismann, 2015): (1) the face value $N$ of a zero bond and (2) a long position in $C_r$ shares generating a payoff only if the CoCo materializes at time $\tau$. This component can be approximated with a knock-in forward. The intuition behind equation 3.12 is that if the share price falls below a certain level $S^*$, an investor will use the face value $N$ to exercise the knock-in forward. That said, the investor is committed to buy the amount of $C_r$ shares for the price of $C_p$ at maturity $T$.

Before maturity the present value of a Zero-Coupon CoCo $V^{zcoco}$ can be determined by adding up the present value of a zero bond $V^{zb}$ and the present value of a knock-in forward $V_t^{kifwd}$. Hereinafter, the components will be explained briefly.

$$V_t^{zcoco} = V_t^{zb} + V_t^{kifwd} \tag{3.13}$$

with

$$V_t^{zb} = N \exp\left[-r(T-t)\right] \tag{3.14}$$

Moreover, the long position in shares at time $t$ can be approximated with the respective closed-form solution of a knock-in forward. (Hull, 2006)

$$\begin{aligned} V_t^{kifwd} = C_r \Bigg[ & S_t \exp\left[-q\left(T-t\right)\right] \left(\frac{S^*}{S_t}\right)^{2\lambda} \Phi\left(y_1\right) \\ & - K \exp\left[-r\left(T-t\right)\right] \left(\frac{S^*}{S_t}\right)^{2\lambda-2} \Phi\left(y_1 - \sigma\sqrt{T-t}\right) \\ & - K \exp\left[-r\left(T-t\right)\right] \Phi\left(-x_1 - \sigma\sqrt{T-t}\right) \\ & + S_t \exp\left[-q\left(T-t\right)\right] \Phi\left(-x_1\right) \Bigg] \end{aligned} \tag{3.15}$$

with

$$\begin{aligned} C_r &= \frac{\alpha N}{C_p} \\ K &= C_p \\ \lambda &= \frac{r - q + \dfrac{\sigma^2}{2}}{\sigma^2} \\ x_1 &= \frac{\log\left(\dfrac{S_t}{S^*}\right)}{\sigma\sqrt{T-t}} + \lambda\sigma\sqrt{T-t} \\ y_1 &= \frac{\log\left(\dfrac{S^*}{S_t}\right)}{\sigma\sqrt{T-t}} + \lambda\sigma\sqrt{T-t} \end{aligned}$$

It is important, however, to recognize that a subtle difference exists between the actual economic payoff of equation 3.12 and its replication with a knock-in forward, since the knock-in forward replicates an economic ownership of shares at maturity $T$. Though, the triggering of a CoCo forces investors to accept the conversion immediately. This could lead to an economic ownership of shares at trigger time $\tau$ and, thus, prior to $T$. Therefore, one could argue that receiving a knock-in forward in the trigger event disregards the dividends which a shareholder would receive in particular when a CoCo triggers early. De Spiegeleer and Schoutens (2011) argue that dividends can be neglected because distressed banks are likely to behave with great restraint when it comes to dividend payments.

### 3.2.2 Step Two - Adding Coupons

As mentioned earlier, the first step excludes coupon payments from the valuation. Yet, in this step we want to include them in order to replicate the exact payout profile of a CoCo. Therefore, we replace the zero bond in equation 3.13 with a straight bond with regular coupon payments $c$. Besides, a third component has to be added which takes into account the foregone coupon payments if the trigger is touched. This can be modeled with a short position in $k$ binary down-and-in calls with maturity $t_i$. Those binary down-and-in calls are knocked in if the trigger $S^*$ is met and hence, offset all future coupon payments.

The price of a straight bond can be determined with:

$$V_t^{sb} = \sum_{i=1}^{T} c_i \exp\left[-r(t_i - t)\right] + N \exp\left[-r(T - t)\right] \tag{3.16}$$

To price the down-and-in calls one might use the formula of Rubinstein and Reiner (1991):

$$\begin{aligned} V_t^{bdic} = \alpha \sum_{i=1}^{k} c_i \exp\left(-rt_i\right) \Bigg[ & \Phi\left(-x_{1i} + \sigma\sqrt{t_i}\right) \\ & + \left(\frac{S^*}{S_t}\right)^{2\lambda-2} \Phi\left(y_{1i} - \sigma\sqrt{t_i}\right)\Bigg] \end{aligned} \tag{3.17}$$

with

$$\begin{aligned} x_{1i} &= \frac{\log\left(\dfrac{S_t}{S^*}\right)}{\sigma\sqrt{t_i}} + \lambda\sigma\sqrt{t_i} \\ y_{1i} &= \frac{\log\left(\dfrac{S^*}{S_t}\right)}{\sigma\sqrt{t_i}} + \lambda\sigma\sqrt{t_i} \\ \lambda &= \frac{r - q + \dfrac{\sigma^2}{2}}{\sigma^2} \end{aligned}$$

To sum up, the theoretical price of a CoCo $V^{ed}$ at time $t$ pursuant the equity derivative approach consists of three components: (1) a straight bond $V^{sb}$, (2) a knock-in-forward $V^{kifwd}$ and (3) a set of binary down-and-in calls $V^{bdic}$:

$$V_t^{ed} = V_t^{sb} + V_t^{kifwd} - V_t^{bdic} \tag{3.18}$$

### 3.2.3 Parameter Classification and Adjustment

| | Description | Usage | Source |
|---|---|---|---|
| $T$ | CoCo maturity | Static input | Term sheet |
| $N$ | CoCo nominal | Static input | Term sheet |
| $c$ | CoCo coupon rate | Static input | Term sheet |
| $\alpha$ | CoCo nominal conversion factor | Static input | Term sheet |
| $S_0$ | Initial share price of the issuer | Dynamic input | Market data |
| $S^*$ | Trigger share price | Static input | Term sheet |
| $C_p$ | CoCo nominal conversion price | Static input | Term sheet |
| $r$ | Risk-free interest rate | Static input | Market data |
| $q$ | Dividend yield | Static input | Market data |
| $\sigma$ | Implied volatility | Static input | Market data |

Table 3.3: Parameter classification of the equity derivative approach (Wilkens and Bethke, 2014)

### 3.2.4 Model Application

| | Value | Comment |
|---|---|---|
| $T$ | 10yrs | Maturity |
| $N$ | 100 | Nominal |
| $c$ | 6.00 | Annual coupon rate |
| $\alpha$ | 1 | Nominal conversion factor |
| $S_0$ | 100 | Initial share price of the bank |
| $S^*$ | 35 | Trigger share price |
| $C_p$ | 65 | Nominal conversion price |
| $r$ | 1.00% | Risk-free interest rate |
| $q$ | 2.00% | Dividend yield |
| $\sigma$ | 30.00% | Implied volatility |

Table 3.4: Parameter specification of equity derivative approach application (Alvemar and Ericson, 2012)

## 3.3 Structural Approach

A third alternative to price CoCos is the structural approach of Pennacchi (2010). The idea has its roots in the seminal work of Merton (1974), which aims to explain a company's default based on the relationship of its assets and liabilities under a standard Black-Scholes setting. Pennacchi (2010)'s approach expands the idea by modeling the stochastic evolution of a bank's balance sheet respectively of its components. In the following, the assets' rate of return process will be explained. Thereafter, we will outline the assumptions of the model regarding the various liabilities a bank issues to refinance itself including deposits, equity and coupon bonds in the form of CoCos. Lastly, a pricing formula will be illustrated.

### 3.3.1 Structural Banking Model

**Bank Assets and Asset-To-Deposit Ratio**

Pennacchi (2010) assumes that a bank holds a portfolio of loans, equities and off-balance sheet positions as assets whose returns follow a jump-diffusion process. The change of this portfolio $A_t$ is determined by the rate of return and the cash in- respectively outflows. In this context, the symbol $*$ is used to point out the change in value of the portfolio which can be quantified by the rate of return, excluding net cashflows. The aforementioned instantaneous rate of return is denoted as $dA_t^*/A_t^*$ and follows a stochastic process as stated below under the risk-neutral probability measure $\mathbb{Q}$:

$$\frac{dA_t^*}{A_t^*} = (r_t - \lambda_t k_t)\, dt + \sigma dz + \left(Y_{q_{t-}} - 1\right) dq_t \tag{3.19}$$

It should be noted that $r_t$ stands for the risk-free interest rate as defined by the Cox et al. (1985) term-structure model which will be discussed shortly. $dz$ is a Brownian motion, whereby $\sigma$ denotes the volatility of returns of the aforementioned asset portfolio. $q_t$ is a Poisson counting process which increases by one whenever a Poisson-distributed event respectively a jump occurs. Hence, the variable $dq_t$ is one whenever such a jump takes place and zero otherwise. The risk-neutral probability that a jump happens is equal to $\lambda_t dt$ where $\lambda_t$ stands for the intensity of the jump process. Variable $Y_{q_{t-}}$ is a i.i.d. random variable drawn from $\ln(Y_{q_{t-}}) \sim \Phi\left(\mu_y, \sigma_y^2\right)$ at time $t$ where $\mu_y$ stands for the mean jump size and $\sigma_y$ denotes the standard deviation of jumps. In case the random variable $Y_{q_{t-}}$ is greater than one, an upward shift in the bank's asset value can be observed. If the value is smaller than one a downward jump takes place. Given that the risk-neutral expected proportional jump $k_t$ is defined as $k_t = E_t^{\mathbb{Q}}\left[Y_{q_{t-}} - 1\right]$, one can determine $k_t$ with the following formula: $k_t = \exp(\mu_y + \frac{1}{2}\sigma_y^2) - 1$. Thus, the risk-neutral expected change in $A^*$ from the jump element $(Y_{q_{t-}} - 1)dq_t$ equals $\lambda_t k_t dt$ in $dt$. To sum up, the value development of a bank's asset portfolio $A_t^*$ follows largely a continuous process. But disruptive jumps may occur as illustrated below in the graph 3.1.

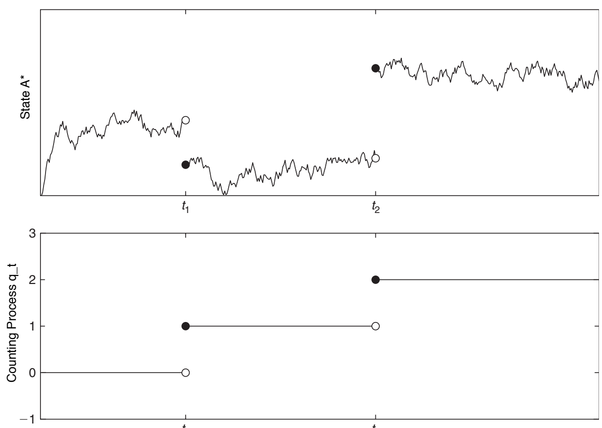


Figure 3.1: The first graph shows two jumps in the state variable $A^*$ at discrete time points. Additionally, the corresponding Poisson counting process $q_t$ is highlighted in the second graph. (Aït-Sahalia and Hansen, 2009)

The risk-neutral process of bank assets $A_t$ including the net cashflows is equal to the assets' rate of return less interest payments $r_t$ respectively premium payments $h_t$ to deposit holders proportionally to their deposits $D_t$. Furthermore, one has to subtract the coupon payments $c_t$ to CoCo investors proportionally to the face value $B$.

$$dA_t = [(r_t - \lambda k) A_t - (r_t + h_t) D_t - c_t B] \, dt + \sigma A_t dz + \left(Y_{q_{t-}} - 1\right) A_t dq \tag{3.20}$$

By substituting variable $x_t$ with $A_t/D_t$ and anticipating the deposit growth process $g(x_t - \hat{x})$ as pointed out by equation 3.31, the risk neutral process of the asset-to-deposit ratio equals:

$$\begin{aligned} \frac{dx_t}{x_t} &= \frac{dA_t}{A_t} - \frac{dD_t}{D_t} \\ &= \left[(r_t - \lambda k) - \frac{r_t + h_t + c_t b_t}{x_t} - g(x_t - \hat{x})\right] dt + \sigma dz + \left(Y_{q_{t-}} - 1\right) dq_t \end{aligned} \tag{3.21}$$

with

$$b_t = \frac{B}{D_t} \tag{3.22}$$

Lastly, an application of Itô's lemma for jump-diffusion processes leads to the following formula for the asset-to-deposit ratio process:

$$\begin{aligned} d \ln(x_t) = &\left[(r_t - \lambda k) - \frac{r_t + h_t + c_t b_t}{x_t} - g(x_t - \hat{x}) - \frac{1}{2}\sigma^2\right] dt \\ &+ \sigma dz + \ln Y_{q_{t-}} dq_t \end{aligned} \tag{3.23}$$

**Default-Free Term Structure**

Pennacchi (2010) applies the term-structure specifications of Cox et al. (1985) to model the risk-neutral process of the instantaneous risk-free interest rate $dr_t$ which is defined as follows:

$$dr_t = \kappa(\bar{r} - r_t) \, dt + \sigma_r \sqrt{r_t} d\zeta \tag{3.24}$$

Note that $\kappa$ is the speed of convergence, $\bar{r}$ is the long-run equilibrium interest rate, $r_t$ is the continuous short-term interest rate, $\sigma_r$ is the instantaneous volatility and $d\zeta$ is a Brownian motion.

A zero bond can be priced using the Cox et al. (1985) specifications under the no-arbitrage assumption. This implies that the price of a risk-free zero bond at time t that pays the amount of €1 in $\tau = T - t$ is given by:

$$P(r_t, \tau) = A(\tau) \exp[-B(\tau) r_t] \tag{3.25}$$

with

$$A(\tau) = \left\{ \frac{2\theta \exp\left[(\theta+\kappa)\frac{\tau}{2}\right]}{(\theta+\kappa)\left[\exp(\theta\tau)-1\right]+2\theta} \right\}^{2\kappa\bar{r}/\sigma_r^2}$$

$$B(\tau) = \frac{2\left[\exp(\theta\tau)-1\right]}{(\theta+\kappa)\left[\exp(\theta\tau)-1\right]+2\theta}$$

$$\theta = \sqrt{\kappa^2 + 2\sigma_r^2}$$

The cost of replication of a risk-free coupon bond that pays a continuous coupon of $c_r dt$ is equal to a set of zero bonds which can be priced with equation 3.25. Therefore, the fair coupon rate $c_r$ of such a coupon bond at time $t$, which is issued at par, equals:

$$\begin{aligned} c_r &= \frac{1 - A(\tau)\exp\left[-B(\tau) r_t\right]}{\int_0^\tau A(s)\exp\left[-B(s) r_t\right] ds} \\ &\approx \frac{1 - A(\tau)\exp\left[-B(\tau) r_t\right]}{\sum_{i=1}^{i=n} A(\Delta t \times i)\exp\left[-B(\Delta t \times i) r_t\right]\Delta t} \end{aligned} \tag{3.26}$$

with

$$n = \frac{\tau}{\Delta t} \tag{3.27}$$

### Deposits and Insurance Premium

Bank deposits are not riskless because depositors may suffer losses if a bank's asset value $A_t$ is worth less than the deposits $D_t$. That said, one can assume that a bank is closed by the deposit insurer when the asset-to-deposit ratio $x_t$ is less or equal to one. A bank might become distressed due to continuous downward movements in its asset value. Then, the bank will be shut down with $A_{t_b} = D_t$ and subsequently, depositors will not face any loss. However, depositors may experience severe losses when a downward jump in asset value happens at a discrete point in time, $\hat{t}$. It may be that the downward jump in asset value exceeds the bank's capital. If such a jump occurs the instantaneous proportional loss to deposits will equal $\left(D_t - Y_{q_{t^-}} A_{\hat{t^-}}\right)/D_t$.

The fair deposit insurance premium $h_t$ for deposit holders can be derived with equation 3.28. The equation illustrates that $h_t$ is closely related to the asset-to-deposit ratio $x_t$:

$$h_t = \lambda\left[\Phi(-d_1) - x_{t^-}\exp\left(\mu_y + \frac{1}{2}\sigma_y^2\right)\Phi(-d_2)\right] \tag{3.28}$$

with

$$d_1 = \frac{\ln(x_{t^-}) + \mu_y}{\sigma_y} \tag{3.29}$$

$$d_2 = d_1 + \sigma_y \tag{3.30}$$

The model assumes that a bank pays continuously a total interest and deposit premium of $(r_t + h_t) D_t dt$ to each depositor. Hence, one can recognize that the deposits of the bank change only because of comparatively higher deposit inflows than outflows. Empirical research of Adrian and Shin (2010) suggests that banks have a target capital ratio and that deposit growth is positively related to the bank's current asset-to deposit ratio:

$$\frac{dD_t}{D_t} = g(x_t - \hat{x})\,dt \tag{3.31}$$

$\hat{x} > 1$ is a bank's target asset-to-deposit ratio with $g$ being a positive constant. Whenever the actual asset-to-deposit ratio is higher than its target, $x_t > \hat{x}$, a bank will shrink its balance sheet. Thus, the deposit growth rate $g(x_t - \hat{x})$ in the time interval $dt$, leads to a mean-reverting tendency for the bank's asset-to-deposit ratio $x_t$.

### Equity and Conversion Threshold

As stated originally, the conversion of a CoCo at time $t_c$ occurs when the asset-to-deposit ratio $x_{t_c}$ meets the trigger level $\bar{x}_{t_c}$. The conversion threshold can also be expressed relative to the original equity-to-deposits ratio $\bar{e}$. This is favourable because the equity value is directly observable in the market whereas the asset value is not. The relationship between the equity threshold $\bar{e}$ and the asset-to-deposit threshold $\bar{x}_{t_c}$ can be summarized as follows:

$$\bar{e} = \frac{E_{t_c}}{D_{t_c}} = \frac{A_{t_c} - D_{t_c} - pB}{D_{t_c}} = \bar{x}_{t_c} - 1 - pb_{t_c} \tag{3.32}$$

Hence, it is possible to specify exactly the conversion trigger of a CoCo bond. This will be important for the valuation part.

### CoCos

The valuation of a CoCo can be accomplished with a Monte Carlo simulation of both the asset and the deposit process. Along the asset-to-deposit ratio process, the CoCo pays coupons and the nominal at maturity unless the CoCo has not been triggered. If the trigger event occurs the conversion amount is paid out. (Wilkens and Bethke, 2014) The price of the CoCo $V^{st}$ is equal to the risk-neutral expectation of the aforementioned cashflows as derived by Pennacchi (2010):

$$V_0^{st} = E_0^{\mathbb{Q}} \left[ \int_0^T \exp \left( - \int_0^t r_s ds \right) v\left(t\right) dt \right] \tag{3.33}$$

Please note that $v(t)$ stands for a CoCo's coupon payment at date $t$ which equals $c_t B$ as long as the CoCo has not been triggered. If the CoCo does not convert until maturity $T$, a final payout of $B$ will be performed. However, if the CoCo triggers early at time $t_c$, there is the one-time cashflow of $pB$. Parameter $p$ determines the maximum conversion amount of new equity per par value of contingent capital. Thereafter, $v(t)$ is zero.

### 3.3.2 Parameter Classification and Adjustment

| | Description | Usage | Source |
|---|---|---|---|
| $T$ | CoCo maturity | Static input | Term sheet |
| $B$ | CoCo nominal | Static input | Term sheet |
| $c$ | CoCo coupon rate | Static input | Term sheet |
| $p$ | CoCo nominal conversion factor | Static input | Term sheet |
| $x_0$ | Initial asset-to-deposit ratio | Dynamic input | Balance sheet |
| $\hat{x}$ | Target asset-to-deposit ratio | Static input | Assumption |
| $g$ | Mean-reversion speed | Static input | Assumption |
| $\sigma$ | Annual asset return volatility | Static input | Assumption |
| $\lambda$ | Jump intensity in asset return process | Static input | Assumption |
| $\mu_y$ | Mean jump size in asset return process | Static input | Assumption |
| $\sigma_y$ | Jump volatility in asset return process | Static input | Assumption |
| $r_0$ | Initial risk-free interest rate | Dynamic input | Market data |
| $\bar{r}$ | Long-term risk-free interest rate | Static input | Market data |
| $\sigma_r$ | Interest rate volatility | Static input | Market data |
| $\kappa$ | Speed of convergence | Static input | Assumption |
| $\rho$ | Correlation between Brownian motion for asset returns and interest rate process | Static input | Market data |
| $\bar{e}$ | Conversion threshold of the market value of original shareholders' equity to deposit value | Static input | Balance sheet |
| $b_0$ | Ratio of the contingent capital's nominal to the initial value of deposits | Dynamic input | Balance sheet |

Table 3.5: Parameter classification of the structural approach (Wilkens and Bethke, 2014)

### 3.3.3 Model Application

| | Value | Comment |
|---|---|---|
| $T$ | 10yrs | Maturity |
| $B$ | 100.00% | Nominal |
| $c$ | 6.00% | Annual coupon rate |
| $p$ | 1 | Nominal conversion factor |
| $x_0$ | 1.15 | Initial asset-to-deposit ratio |
| $\hat{x}$ | 1.1 | Target asset-to-deposit ratio |
| $g$ | 0.5 | Mean-reversion speed |
| $\sigma$ | 2.00% | Annual asset return volatility |
| $\lambda$ | 1 | Jump intensity in asset return process |
| $\mu_y$ | −1.00% | Mean jump size in asset return process |
| $\sigma_y$ | 2.00% | Jump volatility in asset return process |
| $r_0$ | 1.00% | Risk-free interest rate |
| $\bar{r}$ | 6.90% | Long-term risk-free interest rate |
| $\sigma_r$ | 7.00% | Interest rate volatility |
| $\kappa$ | 11.40% | Speed of convergence |
| $\rho$ | −20.00% | Correlation between Brownian motion for asset returns and interest rate process |
| $\bar{e}$ | 2.00% | Conversion threshold of the market value of original shareholders' equity to deposit value |
| $b_0$ | 4.00% | Ratio of contingent capital's nominal to the initial deposit value |

Table 3.6: Parameter specification of structural approach application (Alvemar and Ericson, 2012; Pennacchi, 2010)

# Chapter 4

# Dynamics and Sensitivity Analysis

## 4.1 Credit Derivative Approach

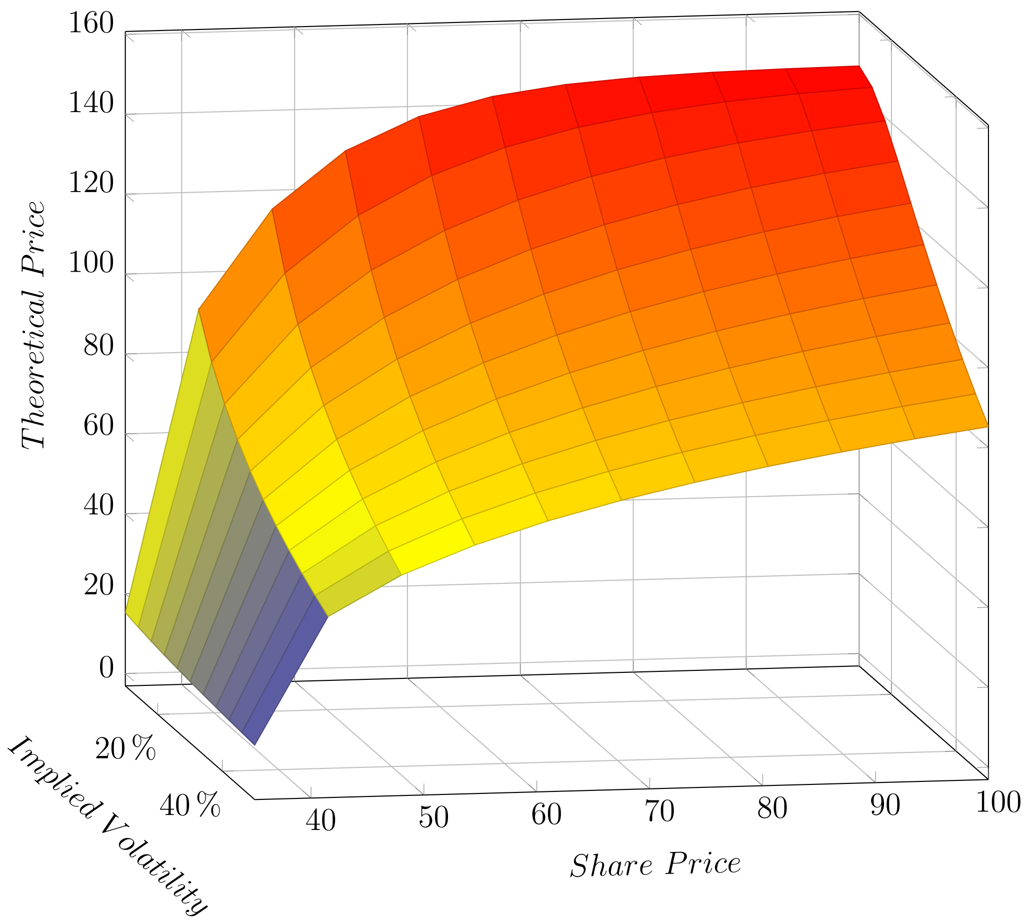


Figure 4.1: CoCo price $V^{cd}$ pursuant to the credit derivative approach (De Spiegeleer and Schoutens, 2011) as function of share price $S$ and implied volatility $\sigma$

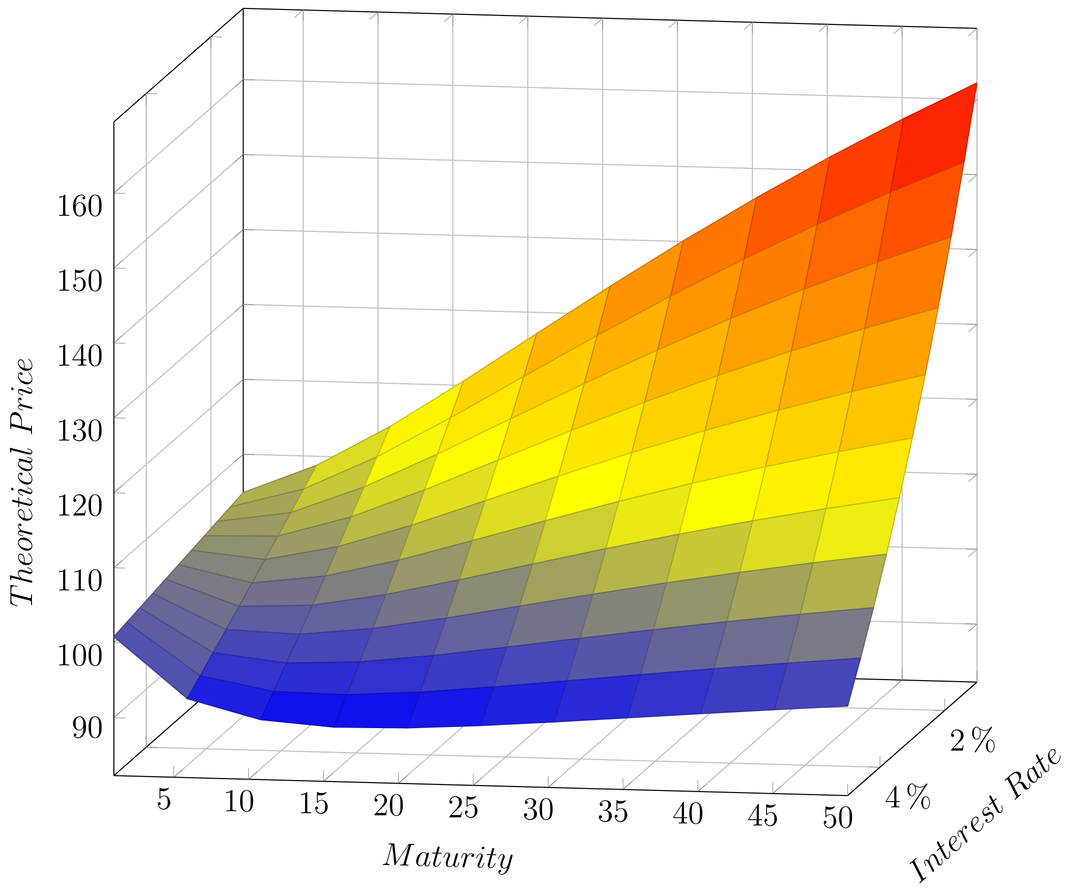


Figure 4.2: CoCo price $V^{cd}$ pursuant to the credit derivative approach (De Spiegeleer and Schoutens, 2011) as function of maturity $T$ and risk-free interest rate $r$

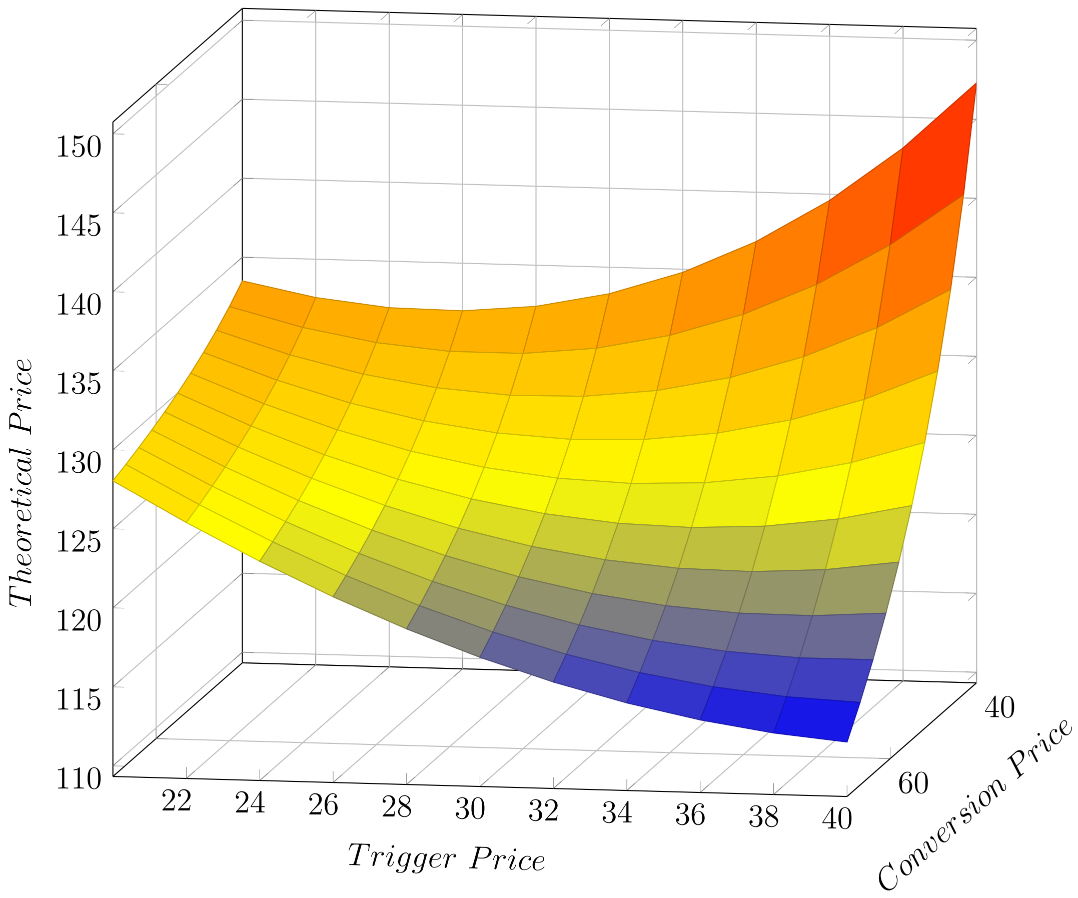


Figure 4.3: CoCo price $V^{cd}$ pursuant to the credit derivative approach (De Spiegeleer and Schoutens, 2011) as function of trigger price $S^*$ and conversion price $C_p$

## 4.2 Equity Derivative Approach

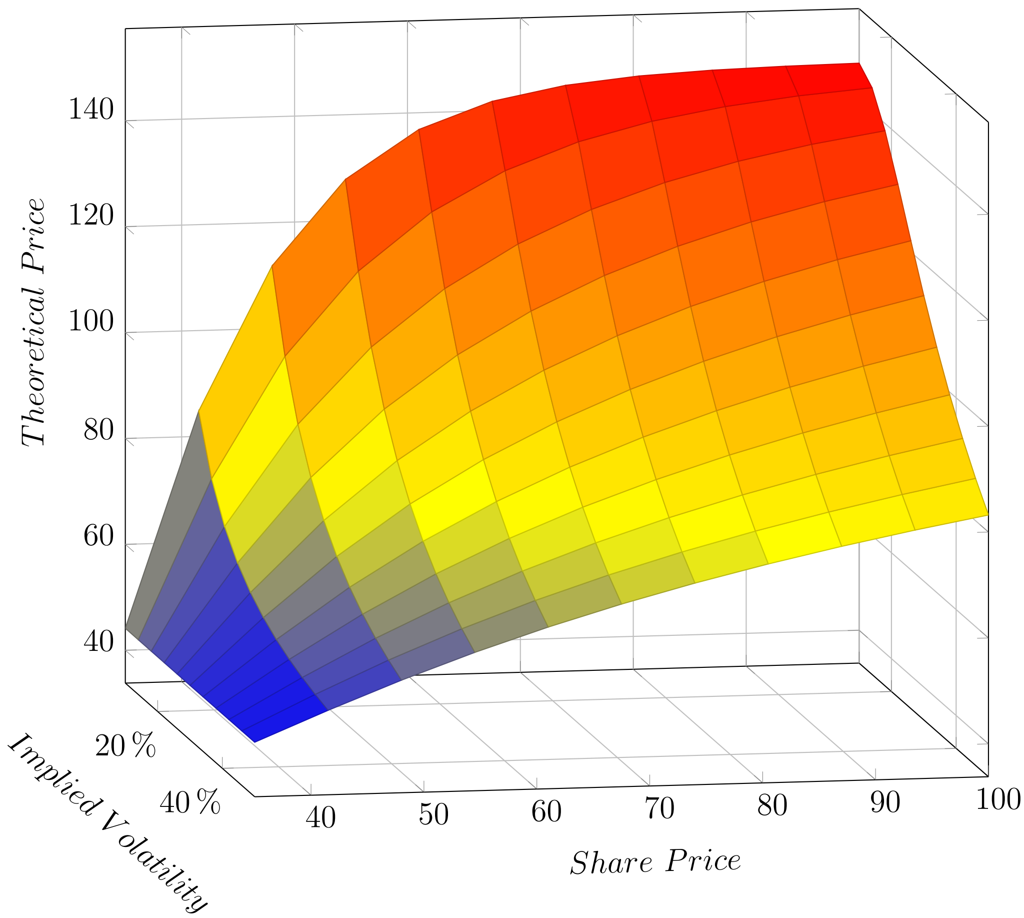


Figure 4.4: CoCo price $V^{ed}$ pursuant to the equity derivative approach (De Spiegeleer and Schoutens, 2011) as function of share price $S$ and volatility $\sigma$

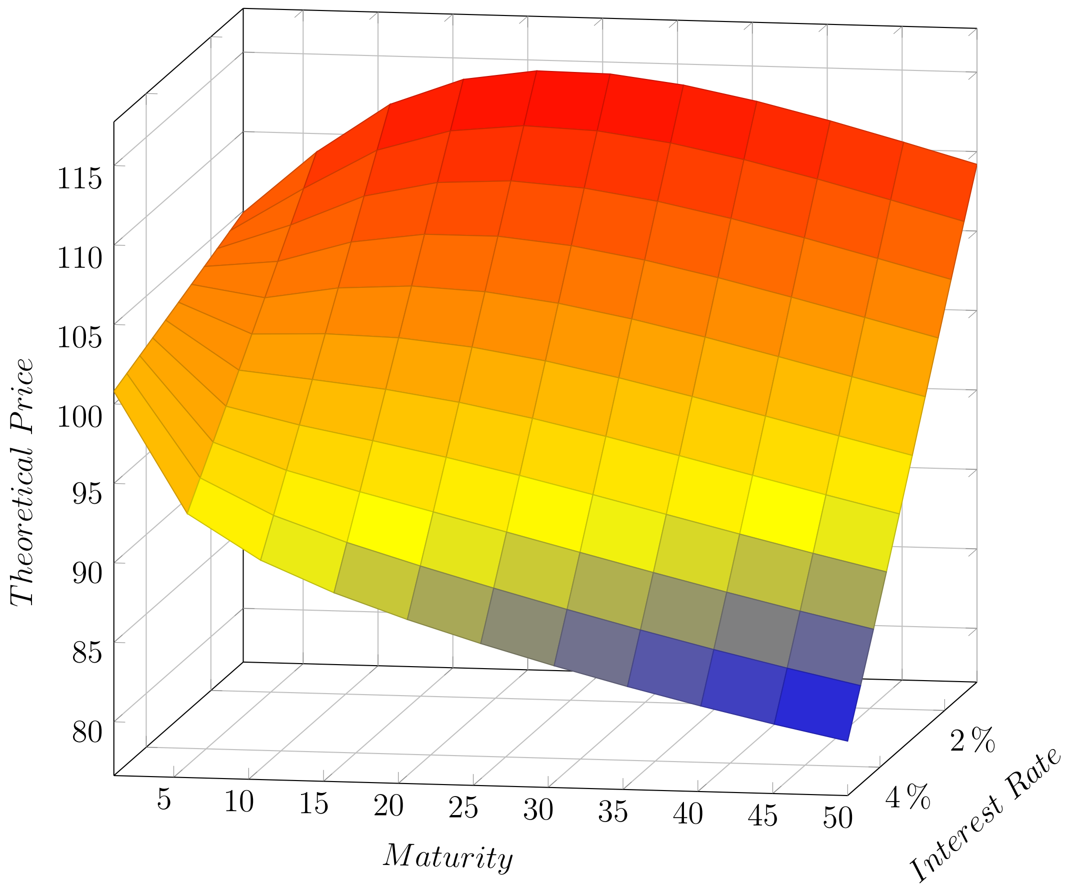


Figure 4.5: CoCo price $V^{ed}$ pursuant to the credit derivative approach (De Spiegeleer and Schoutens, 2011) as function of maturity $T$ and risk-free interest rate $r$

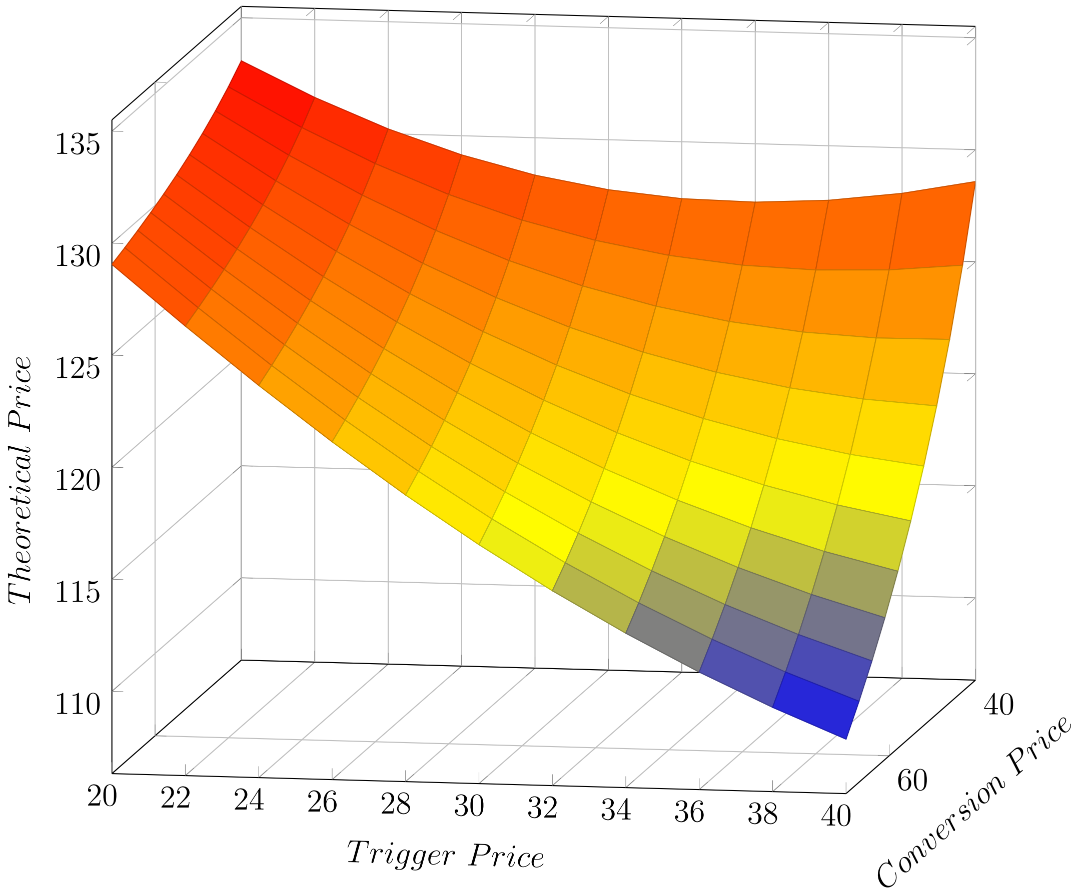


Figure 4.6: CoCo price $V^{ed}$ pursuant to the credit derivative approach (De Spiegeleer and Schoutens, 2011) as function of trigger price $S^*$ and conversion price $C_p$

## 4.3 Structural Approach

[1]

[1] The Monte-Carlo simulation runs in the Amazon Elastic Compute Cloud (EC2) as the service provides a re-sizable compute capacity which is key to quickly scale the computing requirements. If one wants to replicate the simulations it is recommended to follow the instructions of Shekel (2015) to set up a Rstudio server on Amazon EC2.

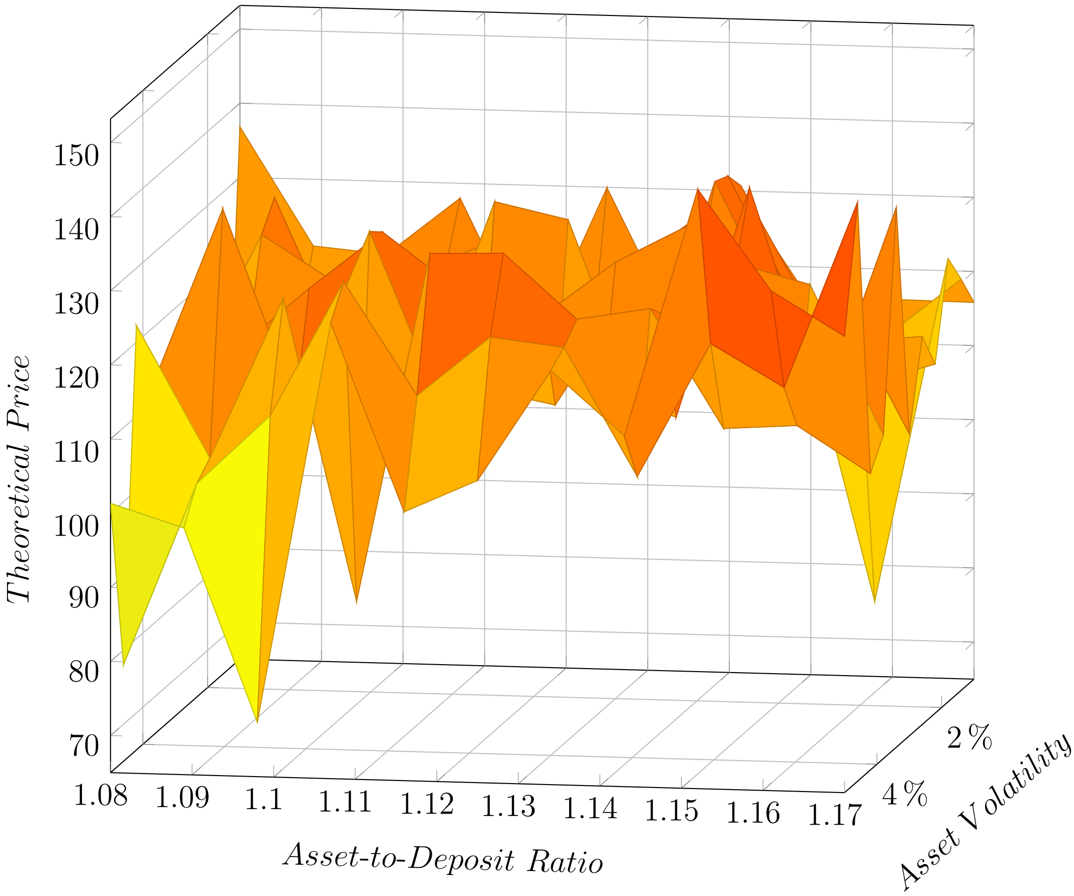


Figure 4.7: CoCo price $V^{sa}$ pursuant to the structural approach (Pennacchi, 2010) as function of initial asset-to-deposit ratio $x_0$ and asset volatility $\sigma$

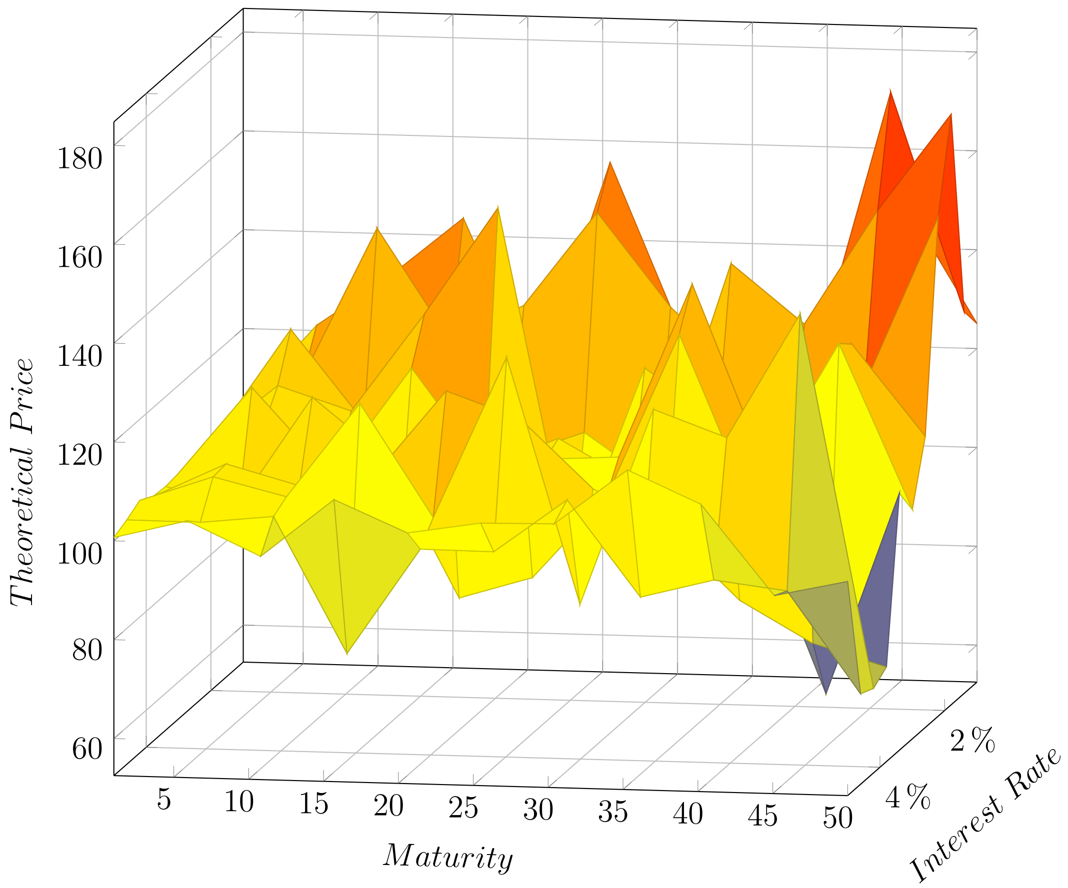


Figure 4.8: CoCo price $V^{sa}$ pursuant to the structural approach (Pennacchi, 2010) as function of maturity $T$ and interest rate $r$

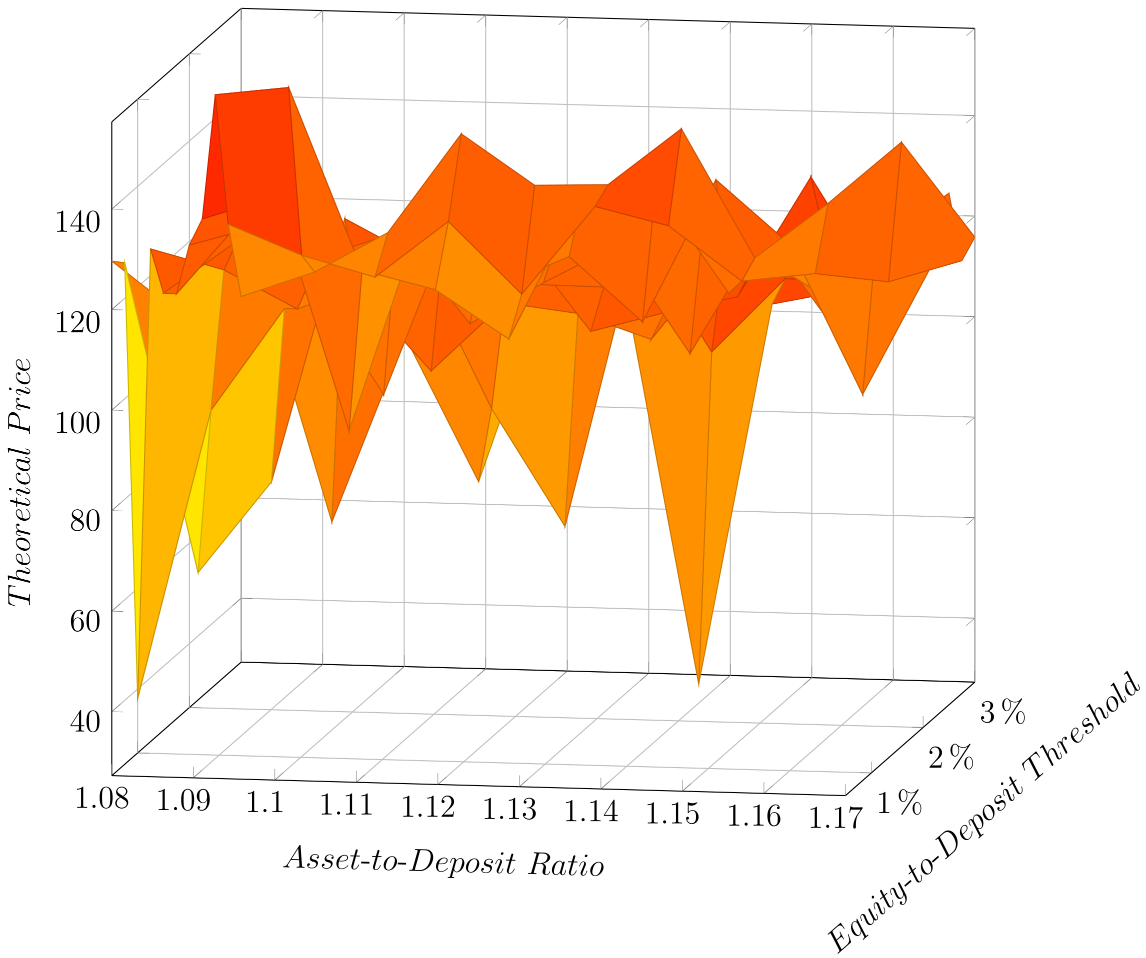


Figure 4.9: CoCo price $V^{sa}$ pursuant to the structural approach (Pennacchi, 2010) as function of initial asset-to-deposit ratio $x_0$ and equity-to-deposit threshold $\bar{e}$

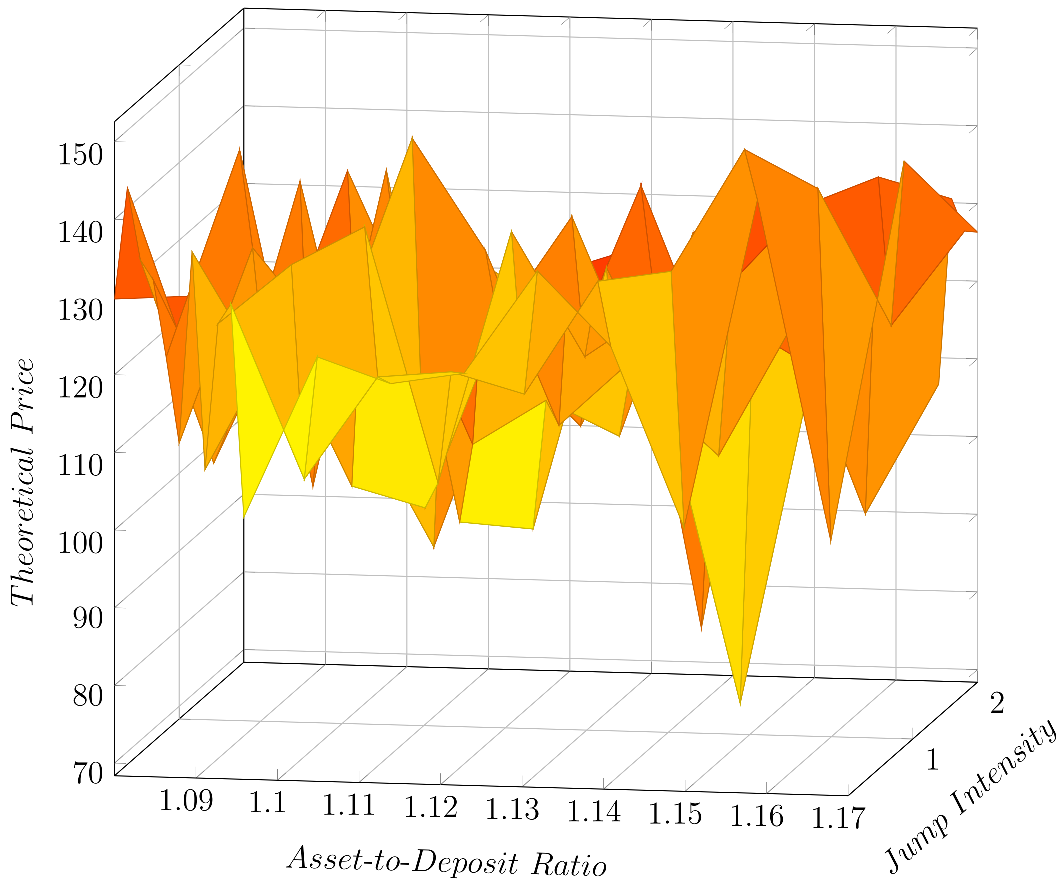


Figure 4.10: CoCo price $V^{sa}$ pursuant to the structural approach (Pennacchi, 2010) as function of initial asset-to-deposit ratio $x_0$ and jump intensity $\lambda$

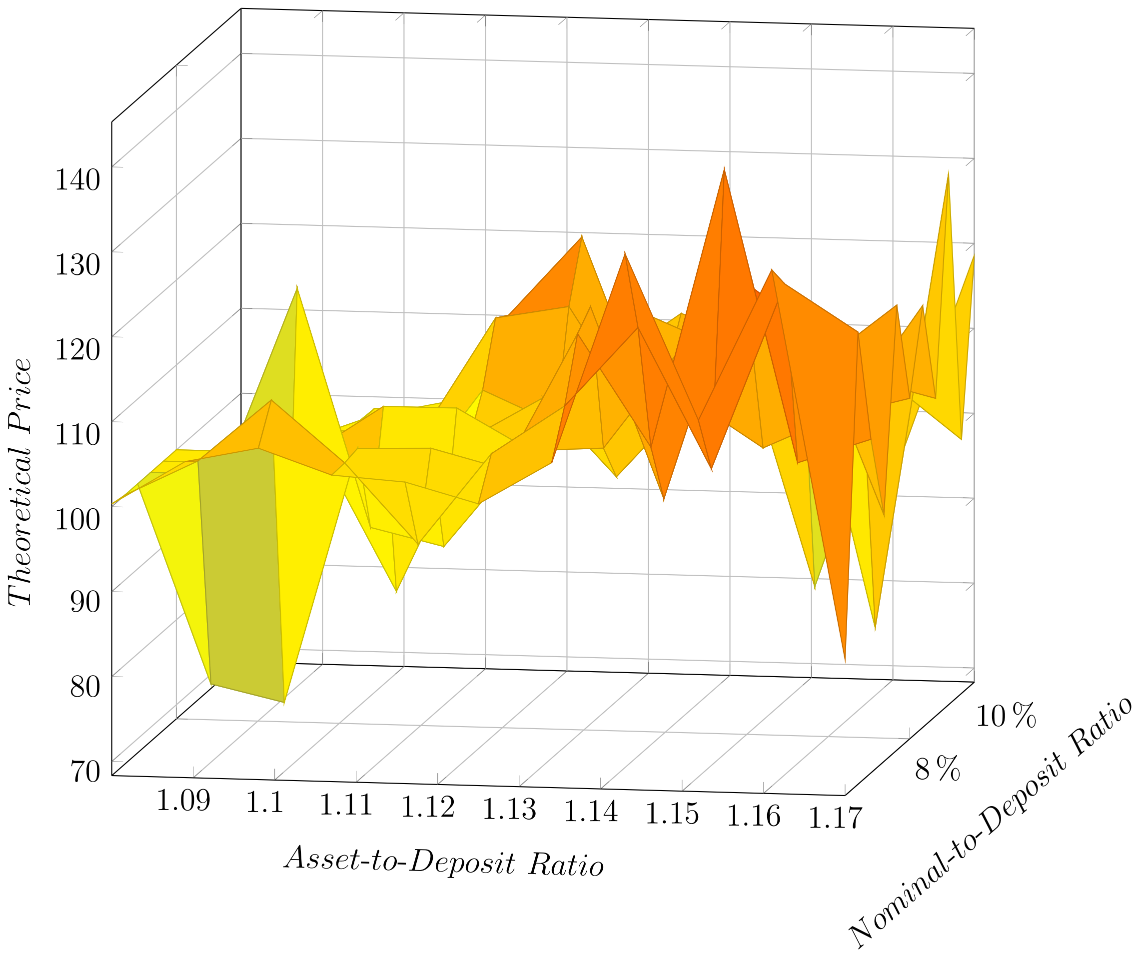


Figure 4.11: CoCo price $V^{sa}$ pursuant to the structural approach (Pennacchi, 2010) as function of initial asset-to-deposit ratio $x_0$ and initial ratio of contingent capital's nominal to the initial value of deposits

# Chapter 5

# Empirical Analysis and Model Comparison

## 5.1 Data Description

### 5.1.1 Deutsche Bank

## 5.2 Model Parametrization

## 5.3 Model Comparison

### 5.3.1 Qualitative Analysis

### 5.3.2 Quantitative Analysis

# Chapter 6

# Conclusion

# Appendix A

# Sample Title

# Appendix B

# Code - Credit Derivative Approach

The following source code is an implementation of the Credit Derivative Approach (De Spiegeleer and Schoutens, 2011) written in R.

```
# Price of Contingent Convertible Bond
price_coco_cd <- function(t, T, S_t, S_star, C_p, c_i, r, N, q, sigma){

  spread_coco <- calc_spread_coco(t, T, S_t, S_star, C_p, r, q, sigma)
  V_t_coco <- N * exp(-(r + spread_coco) * (T - t))

  for (t in 1:T){
    V_t_coco <- V_t_coco + c_i * exp(-(r + spread_coco) * t)
  }
  return(V_t_coco)
}

# Calculation of Trigger Probability
calc_p_star <- function(t, T, S_t, S_star, r, q, sigma){
  p_star <- pnorm((log(S_star / S_t) - calc_mu(r, q, sigma) * (T - t)) /
     (sigma * sqrt(T - t))) + (S_star / S_t)^(2 * calc_mu(r, q, sigma) /
     sigma^2) * pnorm((log(S_star / S_t) + calc_mu(r, q, sigma) * (T - t
    )) / (sigma * sqrt(T - t)))
  return(p_star)
}

# Calculation of Drift of Underlying
calc_mu <- function(r, q, sigma){
  mu <- r - q - sigma^2 / 2
  return(mu)
}

# Spread of CoCo Bond
calc_spread_coco <- function(t, T, S_t, S_star, C_p, r, q, sigma){
  spread_coco <- - log(1 - calc_p_star(t, T, S_t, S_star, r, q, sigma))
    / (T - t) * (1 - S_star / C_p)
  return(spread_coco)
}

# Pricing Example
```

```
price_coco_cd(t <- 0, T <- 5, S_t <- 40, S_star <- 20, C_p <- 25, c_i <-
    7, r <- 0.03, N <- 100, q <- 0, sigma <- 0.3)
```

# Appendix C

# Code - Equity Derivative Approach

The following source code is an implementation of the Equity Derivative Approach (De Spiegeleer and Schoutens, 2011) written in R.

```
# Price of Contingent Convertible Bond
price_coco_ed <- function(t, T, S_t, S_star, C_p, c_i, r, N, q, sigma,
    alpha){
  V_t_ed <- price_cb(t, T, c_i, r, N) - price_dibi(t, T, S_t, S_star, c_
    i, r, q, sigma, alpha) + price_difwd(t, T, S_t, S_star, C_p, r, N, q
    , sigma, alpha)

  return(V_t_ed)
}

# Price of Corporate Bond
price_cb <- function(t, T, c_i, r, N){
  V_t_cb <- N * exp(-r * (T - t))

  for (t in 1:T){
  V_t_cb <- V_t_cb + c_i * exp(-r * t)
  }

  return(V_t_cb)
}

# Price of Binary Option
price_dibi <- function(t, T, S_t, S_star, c_i, r, q, sigma, alpha){
  V_t_dibi <- 0

  i <- t
  k <- T

  for (i in 1:k) {
  V_t_dibi <- V_t_dibi + c_i * exp(- r * i) * (pnorm(- calc_x_1_i(S_t, S
    _star, sigma, r, q, i) + sigma * sqrt(i)) + (S_star / S_t)^(2 * calc
    _lambda(r, q, sigma) - 2) * pnorm ( calc_y_1_i(S_t, S_star, sigma, r
    , q, i) - sigma * sqrt(i)))
  }

```

```r
  V_t_dibi <- alpha * V_t_dibi

  return(V_t_dibi)
}

# Price of Down-And-In Forward
price_difwd <- function(t, T, S_t, S_star, C_p, r, N, q, sigma, alpha){
  V_t_difwd <- calc_conversion_rate(C_p, N, alpha) * (S_t * exp(- q * (T - t)) * (S_star / S_t) ^ (2 * calc_lambda(r, q, sigma)) * pnorm(calc_y_1(t, T, S_t, S_star, r, q, sigma)) - C_p * exp(- r * (T - t)) * (S_star / S_t)^(2 * calc_lambda(r, q, sigma) - 2) * pnorm(calc_y_1(t, T, S_t, S_star, r, q, sigma) - sigma * sqrt(T - t)) - C_p * exp(- r * (T - t)) * pnorm(- calc_x_1(t, T, S_t, S_star, r, q, sigma) + sigma * sqrt(T - t)) + S_t * exp(- q * (T - t)) * pnorm(- calc_x_1(t, T, S_t, S_star, r, q, sigma)))

  return(V_t_difwd)
}

# Calculation of Conversion Rate
calc_conversion_rate <- function(C_p, N, alpha){
  C_r <- alpha * N / C_p

  return(C_r)
}

# Calculation of additional Parameters
calc_x_1_i <- function(S_t, S_star, sigma, r, q, t_i){
  x_1_i <- log(S_t / S_star) / (sigma * sqrt(t_i)) + calc_lambda(r, q, sigma) * sigma * sqrt(t_i)

  return(x_1_i)
}

calc_y_1_i <- function(S_t, S_star, sigma, r, q, t_i){
  y_1_i <- log(S_star / S_t) / (sigma * sqrt(t_i)) + calc_lambda(r, q, sigma) * sigma * sqrt(t_i)

  return(y_1_i)
}

calc_lambda <- function(r, q, sigma){
  lambda <- (r - q + sigma^2 / 2) / sigma^2

  return(lambda)
}

calc_x_1 <- function(t, T, S_t, S_star, r, q, sigma){
  x_1 <- log(S_t / S_star) / (sigma * sqrt(T - t)) + calc_lambda(r, q, sigma) * sigma * sqrt(T - t)

  return(x_1)
}

```

```r
calc_y_1 <- function(t, T, S_t, S_star, r, q, sigma){
  y_1 <- log(S_star / S_t) / (sigma * sqrt(T - t)) + calc_lambda(r, q,
    sigma) * sigma * sqrt(T - t)

  return(y_1)
}

# Pricing Example
price_coco_ed(t <- 0, T <- 5, S_t <- 40, S_star <- 20, C_p <- 25, c_i <-
    7, r <- 0.03, N <- 100, q <- 0, sigma <- 0.3, alpha <- 1)
```

# Appendix D

# Code - Structural Approach

The following source code is an implementation of the Structural Approach (Pennacchi, 2010) with support from Quantnet (2011) in translating the source code of Pennacchi (2010) from GAUSS to R.

```
# Price of Contingent Convertible Bond
price_coco_sa <- function(T , npath , rho , kappa , r_bar, r0, sigma_r,
    mu_Y, sigma_Y, lambda, g, x_hat, b0, p, e_bar, sigma_x, x0_low, x0_
    high, x0_nint, B, c_low, c_high, c_nint){
  n <- T * 250
  dt <- T / n

  result <- sim_corrProcess(T, npath, rho, n, dt)
  dW_1 <- result$dW_1
  dW_2corr <- result$dW_2corr

  r <- sim_interestrate(kappa, r_bar, r0, sigma_r, dW_2corr, n, npath,
    dt)

  V_t_sa <- get_price(npath, n, dt, dW_1, dW_2corr, r, mu_Y, sigma_Y,
    lambda, g, x_hat, b0, p, e_bar, sigma_x, x0_low, x0_high, x0_nint, B
    , c_low, c_high, c_nint) * 100
  return(V_t_sa)
}

sim_corrProcess <- function(T, npath, rho, n, dt){
  vect <- c(1, rho, rho, 1)
  RHO <- matrix(vect, nrow = 2)
  chol_RHO <- t(chol(RHO))

  # Create two Brownian Motions
  dW_1 <- matrix(1, n, npath)
  dW_2 <- matrix(1, n, npath)

  for(j in 1:npath)
  {
    dW_1[ , j] <- rnorm(n) * sqrt(dt)
    dW_2[ , j] <- rnorm(n) * sqrt(dt)
  }
```

```
  # Create Correlated Process based on Brownian Motions using Cholesky-Decomposition
  dW_2corr <- matrix(1, n, npath)
  for(j in 1:npath)
  {
    for(i in 1:n)
    {
      dW_2corr[i, j] <- dW_1[i, j] * chol_RHO[2, 1] + dW_2[i, j] * chol_RHO[2, 2]
    }
  }

  return(list("dW_1" = dW_1, "dW_2corr" = dW_2corr))
}

# Create Interest Rate Process
sim_interestrate <- function(kappa, r_bar, r0, sigma_r, dW_2corr, n, npath, dt){
  r <- matrix(r0, n + 1, npath)

  for(j in 1:npath)
  {
    for(i in 1:n)
    {
      r[i + 1, j] <- r[i, j] + kappa * (r_bar - r[i, j]) * dt + sigma_r * sqrt(r[i, j]) * dW_2corr[i, j]
    }
  }

  return(r)
}

get_price <- function(npath, n, dt, dW_1, dW_2corr, r, mu_Y, sigma_Y, lambda, g, x_hat, b0, p, e_bar, sigma_x, x0_low, x0_high, x0_nint, B, c_low, c_high, c_nint){

  c_fit_matrix <- matrix(0, x0_nint, length(lambda))

  for(w in 1:length(lambda))
  {
    # Create Parametres for Jump Process
    phi <- matrix(rbinom( n%*%npath, 1, dt * lambda[w]), n, npath)
    ln_Y <- matrix(rnorm(n%*%npath, mu_Y, sigma_Y), n, npath)

    b <- matrix(b0, n + 1, npath)
    x_bar0 <- 1 + e_bar + p * b0
    x_bar <- matrix(x_bar0, n + 1, npath)

    h <- matrix(1, n, npath)

    k <- exp(mu_Y + 0.5 * sigma_Y^2) - 1

    c <- seq(c_low, c_high, length = c_nint)
```

```
  x0 <- seq(x0_low, x0_high, length = x0_nint)

  for(l in 1:x0_nint)
  {
    for(m in 1:c_nint)
    {
      x <- matrix(x0[l],n+1,npath)
      ln_x0 <- matrix(log(x0[l]),n+1,npath)
      ln_x <- ln_x0
      binom_c <- matrix(1,n+1,npath)

      for(j in 1:npath)
      {
        for(i in 1:n)
        {
          d_1 <- (ln_x[i, j] + mu_Y) / sigma_Y
          d_2 <- d_1 + sigma_Y

          h[i, j] <- lambda[w] * (pnorm( - d_1) - exp(ln_x[i, j]) * exp(mu_Y + 0.5 * sigma_Y^2) * pnorm(-d_2))

          b[i + 1, j] <- b[i, j] * exp(- g[w] * (exp(ln_x[i, j]) - x_hat) * dt)

          ln_x[i + 1, j] <- ln_x[i, j] + ( (r[i, j] - lambda[w] * k) - (r[i, j] + h[i, j] + c[m] * b[i, j]) / exp(ln_x[i, j]) - g[w] * (exp(ln_x[i, j]) - x_hat) - 0.5 * sigma_x^2) * dt + sigma_x * sqrt(dt) * dW_1[i, j] + ln_Y[i,j] * phi[i, j]

          x[i + 1, j] <- exp(ln_x[i + 1, j])

          x_bar[i + 1, j] <- 1 + e_bar + p * b[i + 1, j]

          if(x[i + 1, j] >= x_bar[i + 1, j] && binom_c[i, j] > 0.5)
          {
            binom_c[i + 1, j] <- 1
          }else
          {
            binom_c[i + 1, j] <- 0
          }
        }
      }

      payments <- matrix(c(rep(c[m] * dt, n - 1), B), n, npath) * binom_c[1:n, ]

      for(j in 1:npath){
        for(i in 2:n){
          if(payments[i, j] == 0 && p * b[sum(binom_c[ , j]) + 1, j] <= x[sum(binom_c[ , j]) + 1, j] - 1 ){
            payments[i, j] <- p * B
            break
          }
```

```
            else if(payments[i, j] == 0 && 0 < x[sum(binom_c[ , j]) + 1, j] - 1 && x[sum(binom_c[ , j]) + 1, j] - 1 < p * b[sum(binom_c[ , j]) + 1, j]){
              payments[i, j] <- (x[sum(binom_c[ , j]) + 1, j] - 1) * B / b[sum(binom_c[ , j]) + 1, j]
              break
            }
            else{
              payments[i, j] <- payments[i, j]
            }
          }
        }
        vec_disc_v <- rep(0, npath)
        for(j in 1:npath)
        {
          disc_v <- 0
          int_r <- 0

          for(i in 1:n)
          {
            int_r <- int_r + r[i, j] * dt
            disc_v <- disc_v + exp(- int_r) * payments[i, j]
          }
          vec_disc_v[j] <- disc_v
        }

        V_t_sa <- mean(vec_disc_v)

        return(V_t_sa)
      }
    }
  }
}

# Pricing Example
price_coco_sa(T <- 10, npath <- 100000, rho <- - 0.2, kappa <- 0.114, r_bar <- 0.069, r0 <- 0.01, sigma_r <- 0.07, mu_Y <- -0.01, sigma_Y <- 0.02, lambda <- c(1), g <- c(0.5), x_hat <- 1.1, b0 <- 0.04, p <- 1, e_bar <- 0.02, sigma_x <- 0.02, x0_low <- 1.15, x0_high <- 1.15, x0_nint <- 10, B <- 1, c_low <- 0.06, c_high <- 0.06, c_nint <- 10)
```

# Appendix E

# Code - Sensitivity Analysis

## E.1 Credit Derivative Approach

The following source code is an implementation of the sensitivity analysis of the Credit Derivative Approach (De Spiegeleer and Schoutens, 2011) written in R.

```
source('CreditDerivativeApproach.R')

# CoCo price V^cd as function of share price S and volatility sigma
createData_CD_S_sigma <- function(S_min, S_max, sigma_min, sigma_max){
  data <- matrix(1, 121, 3)
  counter <- 1
  for(S_increment in seq(from=S_min, to=S_max, by=((S_max-S_min)/10)))
  {
    for(sigma_increment in seq(from=sigma_min, to=sigma_max, by=((sigma_max-sigma_min)/10)))
    {
      data[counter, 1] <- S_increment
      data[counter, 2] <- price_coco_cd(t <- 0, T <- 10, S_t <- S_increment, S_star <- 35, C_p <- 65, c_i <- 6, r <- 0.01, N <- 100, q <- 0.02, sigma <- sigma_increment)
      data[counter, 3] <- sigma_increment
      counter <- counter + 1
    }
  }
  write.table(data, file = "createData_CD_S_sigma.txt", row.names = FALSE, quote=FALSE)
}

# CoCo price V^cd as function of maturity T and risk-free interest rate r
createData_CD_T_r <- function(T_min, T_max, r_min, r_max){
  data <- matrix(1, 121, 3)
  counter <- 1
  for(T_increment in seq(from=T_min, to=T_max, by=((T_max-T_min)/10)))
  {
    for(r_increment in seq(from=r_min, to=r_max, by=((r_max-r_min)/10)))
    {
```

```
      data[counter, 1] <- T_increment
      data[counter, 2] <- price_coco_cd(t <- 0, T <- T_increment, S_t <- 100, S_star <- 35, C_p <- 65, c_i <- 6, r <- r_increment, N <- 100, q <- 0.02, sigma <- 0.3)
      data[counter, 3] <- r_increment
      counter <- counter + 1
    }
  }
  write.table(data, file = "createData_CD_T_r.txt", row.names = FALSE, quote=FALSE)
}

# CoCo price V^cd as function of trigger price S^* and conversion price C_p
createData_CD_Sstar_Cp <- function(S_star_min, S_star_max, C_p_min, C_p_max){
  data <- matrix(1, 121, 3)
  counter <- 1
  for(S_star_increment in seq(from=S_star_min, to=S_star_max, by=((S_star_max-S_star_min)/10)))
  {
    for(C_p_increment in seq(from=C_p_min, to=C_p_max, by=((C_p_max-C_p_min)/10)))
    {
      data[counter, 1] <- S_star_increment
      data[counter, 2] <- price_coco_cd(t <- 0, T <- 10, S_t <- 100, S_star <- S_star_increment, C_p <- C_p_increment, c_i <- 6, r <- 0.01, N <- 100, q <- 0.02, sigma <- 0.3)
      data[counter, 3] <- C_p_increment
      counter <- counter + 1
    }
  }
  write.table(data, file = "createData_CD_Sstar_Cp.txt", row.names = FALSE, quote=FALSE)
}

createData_CD_S_sigma(35.01, 100, 0.1, 0.5)
createData_CD_T_r(1, 50, 0.01, 0.05)
createData_CD_Sstar_Cp(20, 40, 40, 70)
```

## E.2 Equity Derivative Approach

The following source code is an implementation of the sensitivity analysis of the Equity Derivative Approach (De Spiegeleer and Schoutens, 2011) written in R.

```
source('EquityDerivativeApproach.R')

# CoCo price V^ed as function of share price S and volatility sigma
createData_ED_S_sigma <- function(S_min, S_max, sigma_min, sigma_max){
  data <- matrix(1, 121, 3)
  counter <- 1
  for(S_increment in seq(from=S_min, to=S_max, by=((S_max-S_min)/10)))
```

```
  {
    for(sigma_increment in seq(from=sigma_min, to=sigma_max, by=((sigma_max-sigma_min)/10)))
    {
      data[counter, 1] <- S_increment
      data[counter, 2] <- price_coco_ed(t <- 0, T <- 10, S_t <- S_increment, S_star <- 35, C_p <- 65, c_i <- 6, r <- 0.01, N <- 100, q <- 0.02, sigma <- sigma_increment, alpha <- 1)
      data[counter, 3] <- sigma_increment
      counter <- counter + 1
    }
  }
  write.table(data, file = "createData_ED_S_sigma.txt", row.names = FALSE, quote=FALSE)
}

# CoCo price V^ed as function of maturity T and risk-free interest rate r
createData_ED_T_r <- function(T_min, T_max, r_min, r_max){
  data <- matrix(1, 121, 3)
  counter <- 1
  for(T_increment in seq(from=T_min, to=T_max, by=((T_max-T_min)/10)))
  {
    for(r_increment in seq(from=r_min, to=r_max, by=((r_max-r_min)/10)))
    {
      data[counter, 1] <- T_increment
      data[counter, 2] <- price_coco_ed(t <- 0, T <- T_increment, S_t <- 100, S_star <- 35, C_p <- 65, c_i <- 6, r <- r_increment, N <- 100, q <- 0.02, sigma <- 0.3, alpha <- 1)
      data[counter, 3] <- r_increment
      counter <- counter + 1
    }
  }
  write.table(data, file = "createData_ED_T_r.txt", row.names = FALSE, quote=FALSE)
}

# CoCo price V^ed as function of trigger price S^* and conversion price C_p
createData_ED_Sstar_Cp <- function(S_star_min, S_star_max, C_p_min, C_p_max){
  data <- matrix(1, 121, 3)
  counter <- 1
  for(S_star_increment in seq(from=S_star_min, to=S_star_max, by=((S_star_max-S_star_min)/10)))
  {
    for(C_p_increment in seq(from=C_p_min, to=C_p_max, by=((C_p_max-C_p_min)/10)))
    {
      data[counter, 1] <- S_star_increment
      data[counter, 2] <- price_coco_ed(t <- 0, T <- 10, S_t <- 100, S_star <- S_star_increment, C_p <- C_p_increment, c_i <- 6, r <- 0.01, N <- 100, q <- 0.02, sigma <- 0.3, alpha <- 1)
      data[counter, 3] <- C_p_increment
```

```
      counter <- counter + 1
    }
  }
  write.table(data, file = "createData_ED_Sstar_Cp.txt", row.names = FALSE, quote=FALSE)
}

createData_ED_S_sigma(35.01, 100, 0.1, 0.5)
createData_ED_T_r(1, 50, 0.01, 0.05)
createData_ED_Sstar_Cp(20, 40, 40, 70)
```

## E.3 Structural Approach

The following source code is an implementation of the sensitivity analysis of the Structural Approach (Pennacchi, 2010) written in R.

```
source('CreditDerivativeApproach.R')

# CoCo price V^st as function of initial asset-to-deposit ratio x_0 and volatility sigma
createData_SA_x0_sigma <- function(x0_min, x0_max, sigma_min, sigma_max){
  data <- matrix(1, 121, 3)
  counter <- 1
  for(x0_increment in seq(from=x0_min, to=x0_max, by=((x0_max-x0_min)/10)))
  {
    for(sigma_increment in seq(from=sigma_min, to=sigma_max, by=((sigma_max-sigma_min)/10)))
    {
      data[counter, 1] <- x0_increment
      data[counter, 2] <- price_coco_sa(T <- 10, npath <- 5000, rho <- -0.2, kappa <- 0.114, r_bar <- 0.069, r0 <- 0.01, sigma_r <- 0.07, mu_Y <- -0.01, sigma_Y <- 0.02, lambda <- c(1), g <- c(0.5), x_hat <- 1.1, b0 <- 0.04, p <- 1, e_bar <- 0.02, sigma_x <- sigma_increment, x0_low <- x0_increment, x0_high <- x0_increment, x0_nint <- 10, B <- 1, c_low <- 0.06, c_high <- 0.06, c_nint <- 10)
      data[counter, 3] <- sigma_increment
      counter <- counter + 1
    }
  }
  write.table(data, file = "createData_SA_x0_sigma.txt", row.names = FALSE, quote=FALSE)
}

# CoCo price V^sa as function of maturity T and risk-free interest rate r
createData_SA_T_r <- function(T_min, T_max, r_min, r_max){
  data <- matrix(1, 121, 3)
  counter <- 1
  for(T_increment in seq(from=T_min, to=T_max, by=((T_max-T_min)/10)))
  {
```

```r
    for(r_increment in seq(from=r_min, to=r_max, by=((r_max-r_min)/10)))
    {
      data[counter, 1] <- T_increment
      data[counter, 2] <- price_coco_sa(T <- T_increment, npath <- 5000, rho <- - 0.2, kappa <- 0.114, r_bar <- 0.069, r0 <- r_increment, sigma_r <- 0.07, mu_Y <- -0.01, sigma_Y <- 0.02, lambda <- c(1), g <- c(0.5), x_hat <- 1.1, b0 <- 0.04, p <- 1, e_bar <- 0.02, sigma_x <- 0.02, x0_low <- 1.15, x0_high <- 1.15, x0_nint <- 10, B <- 1, c_low <- 0.06, c_high <- 0.06, c_nint <- 10)
      data[counter, 3] <- r_increment
      counter <- counter + 1
    }
  }
  write.table(data, file = "createData_SA_T_r.txt", row.names = FALSE, quote=FALSE)
}

# CoCo price V^sa as function of initial asset-to-deposit ratio x_0 and equity-to-deposit threshold bar_e
createData_SA_x0_ebar <- function(x0_min, x0_max, ebar_min, ebar_max){
  data <- matrix(1, 121, 3)
  counter <- 1
  for(x0_increment in seq(from=x0_min, to=x0_max, by=((x0_max-x0_min)/10)))
  {
    for(ebar_increment in seq(from=ebar_min, to=ebar_max, by=((ebar_max-ebar_min)/10)))
    {
      data[counter, 1] <- x0_increment
      data[counter, 2] <- price_coco_sa(T <- 10, npath <- 5000, rho <- -0.2, kappa <- 0.114, r_bar <- 0.069, r0 <- 0.01, sigma_r <- 0.07, mu_Y <- -0.01, sigma_Y <- 0.02, lambda <- c(1), g <- c(0.5), x_hat <- 1.1, b0 <- 0.04, p <- 1, e_bar <- ebar_increment, sigma_x <- 0.02, x0_low <- x0_increment, x0_high <- x0_increment, x0_nint <- 10, B <- 1, c_low <- 0.06, c_high <- 0.06, c_nint <- 10)
      data[counter, 3] <- ebar_increment
      counter <- counter + 1
    }
  }
  write.table(data, file = "createData_SA_x0_ebar.txt", row.names = FALSE, quote=FALSE)
}

# CoCo price V^st as function of initial asset-to-deposit ratio x_0 and jump intensity in asset return process lambda
createData_SA_x0_lambda <- function(x0_min, x0_max, lambda_min, lambda_max){
  data <- matrix(1, 121, 3)
  counter <- 1
  for(x0_increment in seq(from=x0_min, to=x0_max, by=((x0_max-x0_min)/10)))
  {
    for(lambda_increment in seq(from=lambda_min, to=lambda_max, by=((lambda_max-lambda_min)/10)))
```

```
    {
      data[counter, 1] <- x0_increment
      data[counter, 2] <- price_coco_sa(T <- 10, npath <- 5000, rho <- -0.2, kappa <- 0.114, r_bar <- 0.069, r0 <- 0.01, sigma_r <- 0.07, mu_Y <- -0.01, sigma_Y <- 0.02, lambda <- c(lambda_increment), g <- c(0.5), x_hat <- 1.1, b0 <- 0.04, p <- 1, e_bar <- 0.02, sigma_x <- 0.02, x0_low <- x0_increment, x0_high <- x0_increment, x0_nint <- 10, B <- 1, c_low <- 0.06, c_high <- 0.06, c_nint <- 10)
      data[counter, 3] <- lambda_increment
      counter <- counter + 1
    }
  }
  write.table(data, file = "createData_SA_x0_lambda.txt", row.names = FALSE, quote=FALSE)
}

# CoCo price V^st as function of initial asset-to-deposit ratio x_0 and initial ratio of contingent capital to deposits b0
createData_SA_x0_b0 <- function(x0_min, x0_max, b0_min, b0_max){
  data <- matrix(1, 121, 3)
  counter <- 1
  for(x0_increment in seq(from=x0_min, to=x0_max, by=((x0_max-x0_min)/10)))
  {
    for(b0_increment in seq(from=b0_min, to=b0_max, by=((b0_max-b0_min)/10)))
    {
      data[counter, 1] <- x0_increment
      data[counter, 2] <- price_coco_sa(T <- 10, npath <- 5000, rho <- -0.2, kappa <- 0.114, r_bar <- 0.069, r0 <- 0.01, sigma_r <- 0.07, mu_Y <- -0.01, sigma_Y <- 0.02, lambda <- c(1), g <- c(0.5), x_hat <- 1.1, b0 <- b0_increment, p <- 1, e_bar <- 0.02, sigma_x <- 0.02, x0_low <- x0_increment, x0_high <- x0_increment, x0_nint <- 10, B <- 1, c_low <- 0.06, c_high <- 0.06, c_nint <- 10)
      data[counter, 3] <- b0_increment
      counter <- counter + 1
    }
  }
  write.table(data, file = "createData_SA_x0_b0.txt", row.names = FALSE, quote=FALSE)
}

createData_SA_x0_sigma(1.08, 1.17, 0.01, 0.05)
createData_SA_T_r(1, 50, 0.01, 0.05)
createData_SA_x0_ebar(1.08, 1.17, 0.005, 0.03)
createData_SA_x0_lambda(1.08, 1.17, 0, 2)
createData_SA_x0_b0(1.08, 1.17, 0.1, 0.06)
```

# References


Tobias Adrian and Hyun Song Shin. Liquidity and leverage. *Journal of Financial Intermediation*, 19(3):418–437, 2010.

Yacine Aït-Sahalia and Lars Peter Hansen. *Handbook of Financial Econometrics, Vol 1: Tools and Techniques*, volume 1. Elsevier, 2009.

Boris Albul, Dwight M Jaffee, and Alexei Tchistyi. Contingent convertible bonds and capital structure decisions. *Coleman Fung Risk Management Research Center*, 2010.

Per Alvemar and Philip Ericson. Modelling and pricing contingent convertibles. *University of Gothenburg*, 2012.

Markus PH Buergi. Pricing contingent convertibles: a general framework for application in practice. *Financial Markets and Portfolio Management*, 27(1):31–63, 2013.

John C Cox, Jonathan E Ingersoll Jr, and Stephen A Ross. A theory of the term structure of interest rates. *Econometrica: Journal of the Econometric Society*, pages 385–407, 1985.

Jan De Spiegeleer and Wim Schoutens. Pricing contingent convertibles: a derivatives approach. *Available at SSRN 1795092*, 2011.

Jan De Spiegeleer, Wim Schoutens, and Cynthia Van Hulle. *The Handbook of Hybrid Securities: convertible bonds, coco bonds and bail-in.* John Wiley & Sons, 2014.

Mueller Dietegen. Vom wunder- zum schmuddelkind. *Börsen-Zeitung*, 61:13–14, 2016.

Darrell Duffie and Kenneth J Singleton. Modeling term structures of defaultable bonds. *Review of Financial studies*, 12(4):687–720, 1999.

Marc Erismann. *Pricing Contingent Convertible Capital-A Theoretical and Empirical Analysis of Selected Pricing Models.* PhD thesis, University of St. Gallen, 2015.

Paul Glasserman and Behzad Nouri. Contingent capital with a capital-ratio trigger. *Management Science*, 58(10):1816–1833, 2012.

Roberto Henriques and Saul Doctor. Making cocos work: Structural and pricing considerations for contingent capital securities. *European Credit Research, JP Morgan, February*, 15, 2011.

Jens Hilscher and Alon Raviv. Bank stability and market discipline: The effect of contingent capital on risk taking and default probability. *Journal of Corporate Finance*, 29:542–560, 2014.

John C Hull. *Options, futures, and other derivatives.* Pearson Education, 2006.

Robert A Jarrow and Stuart M Turnbull. Pricing derivatives on financial securities subject to credit risk. *The Journal of Finance*, 50(1):53–85, 1995.

David Lando. *Credit risk modeling: theory and applications.* Princeton University Press, 2009.

Dilip B Madan and Wim Schoutens. Conic coconuts: the pricing of contingent capital notes using conic finance. *Mathematics and Financial Economics*, 4(2):87–106, 2011.

Robert C Merton. On the pricing of corporate debt: The risk structure of interest rates. *The Journal of Finance*, 29(2):449–470, 1974.

George Pennacchi. A structural model of contingent bank capital. Working Paper 1004, Federal Reserve Bank of Cleveland, 2010. URL https://ideas.repec.org/p/fip/fedcwp/1004.html.

Quantnet. Translate gauss code to r code, 2011. URL www.quantnet.com/threads/translate-gauss-code-to-r-code.7784/.

Mark Rubinstein and Eric Reiner. Unscrambling the binary code. *Risk Magazine*, 4 (9):20, 1991.

Wolfgang Schmidt. Credit risk, default models and credit derivatives. Frankfurt School of Finance and Management, 2015.

Liad Shekel. Rstudio server on amazon ec2, 2015. URL r-israel.com/wp-content/uploads/2015/06/Rstudio-in-AWS-16_9.pdf.

Lujing Su and Marc Olivier Rieger. How likely is it to hit a barrier? theoretical and emperical estimates. Technical report, Technical Report Working Paper, 2009.

Suresh Sundaresan and Zhenyu Wang. On the design of contingent capital with a market trigger. *The Journal of Finance*, 70(2):881–920, 2015.

Sascha Wilkens and Nastja Bethke. Contingent convertible (coco) bonds: A first empirical assessment of selected pricing models. *Financial Analysts Journal*, 70(2): 59–77, 2014.