# 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

This thesis is dedicated to my parents for their love and support.

Thank you!

# Abstract

Financial crises have led to higher regulatory standards on the capital adequacy of banks. They 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 (2011b). 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

# Chapter 1

# Introduction and Motivation

# Chapter 2

# Structure of CoCos

## 2.1 Description of CoCos
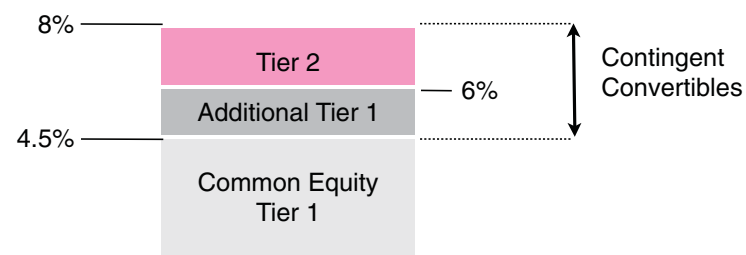


Figure 2.1: CoCos under Basel III (De Spiegeleer and Schoutens, 2011a)

## 2.2 Payoff and Risk Profile
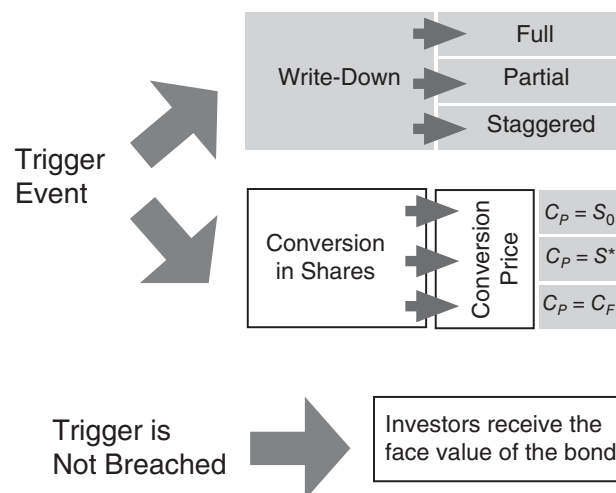


Figure 2.2: Anatomy of CoCos (De Spiegeleer and Schoutens, 2011a)

## 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 $L_{CoCo}$

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

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

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

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

$$P_T = \begin{cases} (1-\alpha)N + C_r S_T^* & \text{if converted} \\ N & \text{if not converted} \end{cases} \tag{2.5}$$

# Chapter 3

# Theory of Pricing

## 3.1 Credit Derivative Approach

The derivation mainly follows De Spiegeleer and Schoutens (2011b). the reduced-form approach which will be the basis of the credit risk approach.

### 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:

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

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

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

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:

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

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) But 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 (2011b) assume that the probability $F^*$, which measures the likelihood that a CoCo triggers within the next 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:

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

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 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:

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

In this context, 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 credit spread $s_{CoCo}$ is driven by a CoCo's major design elements, the trigger level $S^*$ and the conversion price $C_p$:

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

Subsequently, the present value of a CoCo $V^{cd}$ can be calculated by:

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

In summary, the credit derivative approach provides us with a concise method to price CoCos. However, one has to be aware of its largest shortcoming. Losses from cancelled coupon streams are not taken into account. The credit derivative approach naturally overestimates the price of CoCos, but it equips investors with a simple rule of thumb formula.

### 3.1.3  Data Requirements and Calibration

### 3.1.4  Pricing Example

## 3.2  Equity Derivative Approach

Sources: Erismann (2015), De Spiegeleer and Schoutens (2011b)

$$P_T = \mathbb{1}_{\{\tau > T\}} N + \left[ (1 - \alpha) N + \frac{\alpha N}{C_p S^*} \right] \mathbb{1}_{\{\tau \leq T\}}$$

$$= N + \left[ \frac{\alpha N}{C_p} S^* - \alpha N \right] \mathbb{1}_{\{\tau \leq T\}}$$

$$= N + [C_r S^* - \alpha N] \mathbb{1}_{\{\tau \leq T\}}$$

$$= N + C_r \left[ S^* - \frac{\alpha N}{C_r} \right] \mathbb{1}_{\{\tau \leq T\}}$$

$$= N + C_r [S^* - C_p] \mathbb{1}_{\{\tau \leq T\}}$$

$$V_t^{ed} = V_t^{cb} - V_{t_i}^{dibi} + V_t^{difwd} \tag{3.12}$$

## 3.2.1 Corporate Bonds

$$V_t^{cb} = \sum_{i=t}^{T} c_i \exp\left(-rt_i\right) + N \exp\left[-r\left(T-t\right)\right] \tag{3.13}$$

## 3.2.2 Binary Options

$$
\begin{aligned}
V_t^{dibi}\left(c_i, S^*, t\right) = \alpha \sum_{i=1}^{k} c_i \exp\left(-rt_i\right) &\left[ \Phi\left(-x_{1i} + \sigma\sqrt{t_i}\right) \right. \\
&\left. + \left(\frac{S^*}{S_t}\right)^{2\lambda-2} \Phi\left(y_{1i} - \sigma\sqrt{t_i}\right) \right]
\end{aligned}
\tag{3.14}
$$

with

$$x_{1i} = \frac{\log\left(\dfrac{S_t}{S^*}\right)}{\sigma\sqrt{t_i}} + \lambda\sigma\sqrt{t_i}$$

$$y_{1i} = \frac{\log\left(\dfrac{S^*}{S_t}\right)}{\sigma\sqrt{t_i}} + \lambda\sigma\sqrt{t_i}$$

$$\lambda = \frac{r - q + \dfrac{\sigma^2}{2}}{\sigma^2}$$

## 3.2.3 Down-And-In Forward

$$\max\left(S_T - K\right) \ \text{if} \ \min_{0 \le t \le T}\left(S_T\right) \le S^* \tag{3.15}$$

$$\max\left(K - S_T\right) \ \text{if} \ \min_{0 \le t \le T}\left(S_T\right) \le S^* \tag{3.16}$$

$$
\begin{aligned}
V_t^{dic}\left(S_t, S^*, K\right) = S_t \exp\left[-q\left(T-t\right)\right] \left(\frac{S^*}{S_t}\right)^{2\lambda} \Phi\left(y\right) \\
- K \exp\left[-r\left(T-t\right)\right] \left(\frac{S^*}{S_t}\right)^{2\lambda-2} \Phi\left(y - \sigma\sqrt{T-t}\right)
\end{aligned}
\tag{3.17}
$$

with

$$K = C_p$$

$$y = \frac{\log\left(\dfrac{S^{*2}}{S_t K}\right)}{\sigma\sqrt{T-t}} + \lambda\sigma\sqrt{T-t}$$

$$\lambda = \frac{r - q + \dfrac{\sigma^2}{2}}{\sigma^2}$$

$$
\begin{aligned}
V_t^{dip}\left(S_t, S^*, K\right) &= S_t \exp\left[-q\left(T-t\right)\right]\left(\frac{S^*}{S_t}\right)^{2\lambda}\left[\Phi\left(y\right) - \Phi\left(y_1\right)\right] \\
&- K\exp\left[-r\left(T-t\right)\right]\left(\frac{S^*}{S_t}\right)^{2\lambda-2}\left[\Phi\left(y - \sigma\sqrt{T-t}\right) - \Phi\left(y_1 - \sigma\sqrt{T}\right)\right] \\
&+ K\exp\left[-r\left(T-t\right)\right]\Phi\left(x_1 + \sigma\sqrt{T-t}\right) \\
&- S_t\exp\left[-q\left(T-t\right)\right]\Phi\left(-x_1\right)
\end{aligned}
\tag{3.18}
$$

with

$$x_1 = \frac{\log\left(\dfrac{S_t}{S^*}\right)}{\sigma\sqrt{T-t}} + \lambda\sigma\sqrt{T-t}$$

$$y_1 = \frac{\log\left(\dfrac{S^*}{S_t}\right)}{\sigma\sqrt{T-t}} + \lambda\sigma\sqrt{T-t}$$

$$\min\left(S_t\right) \leq S^* : P_T = S_T - K = \max\left(S_T - K\right) - \max\left(K - S_T\right) \tag{3.19}$$

$$\min\left(S_t\right) > S^* : P_T = 0 \tag{3.20}$$

$$V_t^{difwd} = C_r \left[ S_t \exp\left[-q\left(T-t\right)\right] \left(\frac{S^*}{S_t}\right)^{2\lambda} \Phi\left(y_1\right) \right.$$

$$- K \exp\left[-r\left(T-t\right)\right] \left(\frac{S^*}{S_t}\right)^{2\lambda-2} \Phi\left(y_1 - \sigma\sqrt{T-t}\right)$$

$$- K \exp\left[-r\left(T-t\right)\right] \Phi\left(-x_1 - \sigma\sqrt{T-t}\right)$$

$$\left. + S_t \exp\left[-q\left(T-t\right)\right] \Phi\left(-x_1\right) \right] \tag{3.21}$$

with

$$C_r = \frac{\alpha N}{C_p} \tag{3.22}$$

### 3.2.4 Data Requirements and Calibration

### 3.2.5 Pricing Example

## 3.3 Structural Approach

"Structural credit pricing models are based on modeling the stochastic evolution of the balance sheet of the issuer, with default when the issuer is unable or unwilling to meet its obligations." (Duffie and Singleton, 2003)

### 3.3.1 Synthetic Balance Sheet

### 3.3.2 Data Requirements and Calibration

### 3.3.3 Pricing Example

# Chapter 4

# Dynamics and Sensitivity Analysis

# 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, 2011b) written in R.

```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)
  }
  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)))
  p_star
}

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

# Pricing Example
```

```
32 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, 2011b) written in R.

```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)))
  }

}
```

16

```r
30    V_t_dibi <- alpha * V_t_dibi
31
32    return(V_t_dibi)
33  }
34
35  # Price of Down-And-In Forward
36  price_difwd <- function(t, T, S_t, S_star, C_p, r, N, q, sigma, alpha){
37    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)))
38
39    return(V_t_difwd)
40  }
41
42  # Calculation of Conversion Rate
43  calc_conversion_rate <- function(C_p, N, alpha){
44    C_r <- alpha * N / C_p
45
46    return(C_r)
47  }
48
49  # Calculation of additional Parameters
50  calc_x_1_i <- function(S_t, S_star, sigma, r, q, t_i){
51    x_1_i <- log(S_t / S_star) / (sigma * sqrt(t_i)) + calc_lambda(r, q,
        sigma) * sigma * sqrt(t_i)
52
53    return(x_1_i)
54  }
55
56  calc_y_1_i <- function(S_t, S_star, sigma, r, q, t_i){
57    y_1_i <- log(S_star / S_t) / (sigma * sqrt(t_i)) + calc_lambda(r, q,
        sigma) * sigma * sqrt(t_i)
58
59    return(y_1_i)
60  }
61
62  calc_lambda <- function(r, q, sigma){
63    lambda <- (r - q + sigma^2 / 2) / sigma^2
64
65    return(lambda)
66  }
67
68  calc_x_1 <- function(t, T, S_t, S_star, r, q, sigma){
69    x_1 <- log(S_t / S_star) / (sigma * sqrt(T - t)) + calc_lambda(r, q,
        sigma) * sigma * sqrt(T - t)
70
71    return(x_1)
72  }
73
```

```r
74 calc_y_1 <- function(t, T, S_t, S_star, r, q, sigma){
75   y_1 <- log(S_star / S_t) / (sigma * sqrt(T - t)) + calc_lambda(r, q,
       sigma) * sigma * sqrt(T - t)
76
77   return(y_1)
78 }
79
80 # Pricing Example
81 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.

```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)
  }
```

```r
30
31    # Create Correlated Process based on Brownian Motions using Cholesky-
          Decomposition
32    dW_2corr <- matrix(1, n, npath)
33    for(j in 1:npath)
34    {
35      for(i in 1:n)
36      {
37        dW_2corr[i, j] <- dW_1[i, j] * chol_RHO[2, 1] + dW_2[i, j] * chol_
    RHO[2, 2]
38      }
39    }
40
41    return(list("dW_1" = dW_1, "dW_2corr" = dW_2corr))
42 }
43
44 # Create Interest Rate Process
45 sim_interestrate <- function(kappa, r_bar, r0, sigma_r, dW_2corr, n,
          npath, dt){
46    r <- matrix(r0, n + 1, npath)
47
48    for(j in 1:npath)
49    {
50      for(i in 1:n)
51      {
52        r[i + 1, j] <- r[i, j] + kappa * (r_bar - r[i, j]) * dt + sigma_r
    * sqrt(r[i, j]) * dW_2corr[i, j]
53      }
54    }
55
56    return(r)
57 }
58
59 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){
60
61    c_fit_matrix <- matrix(0, x0_nint, length(lambda))
62
63    for(w in 1:length(lambda))
64    {
65      # Create Parametres for Jump Process
66      phi <- matrix(rbinom( n%*%npath, 1, dt * lambda[w]), n, npath)
67      ln_Y <- matrix(rnorm(n%*%npath, mu_Y, sigma_Y), n, npath)
68
69      b <- matrix(b0, n + 1, npath)
70      x_bar0 <- 1 + e_bar + p * b0
71      x_bar <- matrix(x_bar0, n + 1, npath)
72
73      h <- matrix(1, n, npath)
74
75      k <- exp(mu_Y + 0.5 * sigma_Y^2) - 1
76
77      c <- seq(c_low, c_high, length = c_nint)
```

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

79

80      for(l in 1:x0_nint)
81      {
82        for(m in 1:c_nint)
83        {
84          x <- matrix(x0[l],n+1,npath)
85          ln_x0 <- matrix(log(x0[l]),n+1,npath)
86          ln_x <- ln_x0
87          binom_c <- matrix(1,n+1,npath)

88

89          for(j in 1:npath)
90          {
91            for(i in 1:n)
92            {
93              d_1 <- (ln_x[i, j] + mu_Y) / sigma_Y
94              d_2 <- d_1 + sigma_Y

95

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

97

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

99

100             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]

101

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

103

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

105

106             if(x[i + 1, j] >= x_bar[i + 1, j] && binom_c[i, j] > 0.5)
107             {
108               binom_c[i + 1, j] <- 1
109             }else
110             {
111               binom_c[i + 1, j] <- 0
112             }
113           }
114         }

115

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

117

118         for(j in 1:npath){
119           for(i in 2:n){
120             if(payments[i, j] == 0 && p * b[sum(binom_c[ , j]) + 1, j]
    <= x[sum(binom_c[ , j]) + 1, j] - 1 ){
121               payments[i, j] <- p * B
122               break
123             }
```

```r
124            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]){
125              payments[i, j] <- (x[sum(binom_c[ , j]) + 1, j] - 1) * B /
      b[sum(binom_c[ , j]) + 1, j]
126              break
127            }
128            else{
129              payments[i, j] <- payments[i, j]
130            }
131          }
132        }
133      vec_disc_v <- rep(0, npath)
134      for(j in 1:npath)
135      {
136        disc_v <- 0
137        int_r <- 0
138
139        for(i in 1:n)
140        {
141          int_r <- int_r + r[i, j] * dt
142          disc_v <- disc_v + exp(- int_r) * payments[i, j]
143        }
144        vec_disc_v[j] <- disc_v
145      }
146
147      V_t_sa <- mean(vec_disc_v)
148
149      return(V_t_sa)
150    }
151   }
152  }
153 }
154
155 # Pricing Example
156 price_coco_sa(T = 5, npath = 2, rho = - 0.2, kappa = 0.114, r_bar =
      0.069, r0 = 0.035, 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 = 0.05, c_high = 0.05, c_nint = 10)
```

# References

Jan De Spiegeleer and Wim Schoutens. *The handbook of convertible bonds: pricing, strategies and risk management*, volume 581. John Wiley & Sons, 2011a.

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

Darrell Duffie and Kenneth J Singleton. *Credit risk pricing, measurement, and management*. Princeton University Press, 2003.

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

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

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 https://www.quantnet.com/threads/translate-gauss-code-to-r-code.7784/.

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

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.