

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!

Acknowledgements

plenty of waffle, plenty of waffle, plenty of waffle, plenty of waffle, plenty
of waffle, plenty of waffle, plenty of waffle, plenty of waffle.

Abstract

plenty of waffle, plenty of waffle, plenty of waffle, plenty of waffle, plenty
of waffle, plenty of waffle, plenty of waffle, plenty of waffle.

Contents

1	Introduction and Motivation	1
1.1	Introduction	1
1.2	Literature Overview	1
1.3	Motivation	1
1.4	Methodology	1
2	Structure of CoCos	2
2.1	Description of CoCos	2
2.2	Payoff and Risk Profile	2
2.3	Conversion Trigger	2
2.3.1	Market Trigger	2
2.3.2	Accounting Trigger	2
2.3.3	Regulatory Trigger	2
2.3.4	Multivariate Trigger	2
2.4	Conversion Details	2
2.4.1	Conversion Fraction	2
2.4.2	Conversion Price and Ratio	3
3	Theory of Pricing	4
3.1	Credit Derivative Approach	4
3.1.1	Intensity-based Approach	4
3.1.2	Application to CoCos	5
3.1.3	Data Requirements and Calibration	5
3.1.4	Pricing Example	5
3.2	Equity Derivative Approach	5
3.2.1	Corporate Bonds	5
3.2.2	Binary Options	5
3.2.3	Down-And-In Forward	6

3.2.4	Data Requirements and Calibration	7
3.2.5	Pricing Example	7
4	Dynamics and Sensitivity Analysis	8
4.1	Credit Derivative Approach	8
4.2	Equity Derivative Approach	8
5	Empirical Analysis and Model Comparison	9
5.1	Data Description	9
5.1.1	Deutsche Bank	9
5.2	Model Parametrization	9
5.3	Model Comparison	9
5.3.1	Qualitative Analysis	9
5.3.2	Quantitative Analysis	9
6	Conclusion	10
A	Sample Title	11
B	Sample Title	12

List of Figures

Chapter 1

Introduction and Motivation

1.1 Introduction

1.2 Literature Overview

1.3 Motivation

1.4 Methodology

Chapter 2

Structure of CoCos

2.1 Description of CoCos

2.2 Payoff and Risk Profile

2.3 Conversion Trigger

2.3.1 Market Trigger

2.3.2 Accounting Trigger

2.3.3 Regulatory Trigger

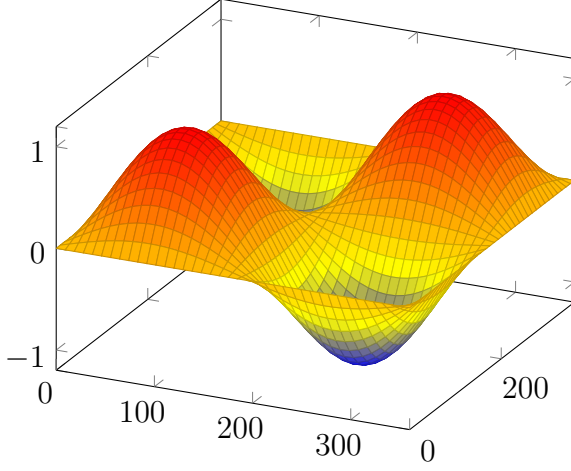
2.3.4 Multivariate Trigger

2.4 Conversion Details

2.4.1 Conversion Fraction

- conversion fraction α
- 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}

$$C_p = \frac{\alpha N}{C_r} \quad (2.1)$$

$$C_r = \frac{\alpha N}{C_p} \quad (2.2)$$

$$R_{CoCo} = \frac{S_T^*}{C_p} \quad (2.3)$$

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

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

Chapter 3

Theory of Pricing

3.1 Credit Derivative Approach

3.1.1 Intensity-based Approach

$$p^* = 1 - \exp(-\lambda_{Trigger} \times T) \quad (3.1)$$

$$s_{CoCo} = (1 - R_{CoCo}) \times \lambda_{Trigger} = Loss_{CoCo} \times \lambda_{Trigger} \quad (3.2)$$

$$Loss_{CoCo} = N - C_r \times S^* = N \left(1 - \frac{S^*}{C_P}\right) \quad (3.3)$$

$$R_{CoCo} = \frac{S^*}{C_p} \quad (3.4)$$

$$p^* = \Phi \left(\frac{\log \left(\frac{S^*}{S} \right) - \mu T}{\sigma \sqrt{T}} \right) + \left(\frac{S^*}{S} \right)^{\frac{2\mu}{\sigma^2}} \Phi \left(\frac{\log \left(\frac{S^*}{S} \right) + \mu T}{\sigma \sqrt{T}} \right) \quad (3.5)$$

$$\lambda_{Trigger} = -\frac{\log(1 - p^*)}{T} \quad (3.6)$$

$$s_{CoCo} = -\frac{\log(1 - p^*)}{T} \times \left(1 - \frac{S^*}{C_p}\right) \quad (3.7)$$

3.1.2 Application to CoCos

3.1.3 Data Requirements and Calibration

3.1.4 Pricing Example

3.2 Equity Derivative Approach

Sources: [?], [?]

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

$$V_t^{ed} = V_t^{cb} - V_{t_i}^{dibi} + V_t^{difwd} \quad (3.8)$$

3.2.1 Corporate Bonds

$$V_t^{cb} = \sum_{i=t}^T c_i \exp(-rt_i) + N \exp[-r(T-t)] \quad (3.9)$$

3.2.2 Binary Options

$$\begin{aligned} V_t^{dibi}(c_i, S^*, t) &= \alpha \sum_{i=t}^T c_i \exp(-rt_i) \left[\Phi(-x_{1i} + \sigma\sqrt{t_i}) \right. \\ &\quad \left. + \left(\frac{S^*}{S} \right)^{2\lambda-2} \Phi(y_{1i} - \sigma\sqrt{t_i}) \right] \end{aligned} \quad (3.10)$$

with

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

3.2.3 Down-And-In Forward

$$\max(S_T - K) \text{ if } \min_{0 \leq t \leq T} (S_T) \leq S^* \quad (3.11)$$

$$\max(K - S_T) \text{ if } \min_{0 \leq t \leq T} (S_T) \leq S^* \quad (3.12)$$

$$\begin{aligned}
V_t^{dic}(S_t, S^*, K) &= S_t \exp[-q(T-t)] \left(\frac{S^*}{S_t}\right)^{2\lambda} \Phi(y) \\
&\quad - K \exp[-r(T-t)] \left(\frac{S^*}{S_t}\right)^{2\lambda-2} \Phi(y - \sigma\sqrt{T-t})
\end{aligned} \quad (3.13)$$

with

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

$$\begin{aligned}
V_t^{dip}(S_t, S^*, K) &= S_t \exp[-q(T-t)] \left(\frac{S^*}{S_t}\right)^{2\lambda} [\Phi(y) - \Phi(y_1)] \\
&\quad - K \exp[-r(T-t)] \left(\frac{S^*}{S_t}\right)^{2\lambda-2} \left[\Phi(y - \sigma\sqrt{T-t}) - \Phi(y_1 - \sigma\sqrt{T-t})\right] \\
&\quad + K \exp[-r(T-t)] \Phi(x_1 + \sigma\sqrt{T-t}) \\
&\quad - S_t \exp[-q(T-t)] \Phi(-x_1)
\end{aligned} \quad (3.14)$$

with

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

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

$$\min(S_t) \leq S^* : P_T = S_T - K = \max(S_T - K) - \max(K - S_T) \quad (3.15)$$

$$\min(S_t) > S^* : P_T = 0 \quad (3.16)$$

$$V_t^{difwd} = C_r \left[S_t \exp[-q(T-t)] \left(\frac{S^*}{S_t}\right)^{2\lambda} \Phi(y_1) \right. \\ \left. - K \exp[-r(T-t)] \left(\frac{S^*}{S}\right)^{2\lambda-2} \Phi(y_1 - \sigma\sqrt{T-t}) \right. \\ \left. - K \exp[-r(T-t)] \Phi(-x_1 - \sigma\sqrt{T-t}) \right. \\ \left. + S_t \exp[-q(T-t)] \Phi(-x_1) \right] \quad (3.17)$$

with

$$C_r = \frac{\alpha N}{C_p} \quad (3.18)$$

3.2.4 Data Requirements and Calibration

3.2.5 Pricing Example

Chapter 4

Dynamics and Sensitivity Analysis

4.1 Credit Derivative Approach

4.2 Equity Derivative Approach

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

Sample Title

The next code will be directly imported from a file

```
1 library(magrittr)
2
3 ### Plain-Vanilla Black-Scholes
4
5 bs <- function(spot, strike, r_d, r_f, sigma, maturity, callput = 1) {
6   forward <- spot * exp((r_d - r_f) * maturity)
7   d_plus <- (log(forward / strike) + sigma ** 2 / 2 * maturity) / (sigma
8     * sqrt(maturity))
9   d_minus <- (log(forward / strike) - sigma ** 2 / 2 * maturity) / (
10     sigma * sqrt(maturity))
11   exp(-r_d * maturity) * callput * (forward * pnorm(callput * d_plus) -
12     strike * pnorm(callput * d_minus))
13 }
14
15 plain_vanilla_call <- function(spot, strike, r_d, r_f, sigma, maturity) {
16   {
17     bs(spot, strike, r_d, r_f, sigma, maturity, 1)
18   }
19 }
20
21 plain_vanilla_put <- function(spot, strike, r_d, r_f, sigma, maturity) {
22   bs(spot, strike, r_d, r_f, sigma, maturity, -1)
23 }
24
25 ### Double Knock-Out Options
26 # Pricing Options with curved boundaries, Naoto Kunitomo, Masayuki Ikeda
27 # (1992)
28 # case for delta_1=delta_2=0
29
30 dko <- function(spot, strike, upper, lower, r_d, r_f, sigma, t, callput
31   = 1) {
32   stopifnot(callput %in% c(-1,1))
33   if (or(spot < lower, spot > upper)) { return(0) }
34   r_prime <- (r_d - r_f + sigma ** 2 / 2) * t
35   sigma_scaled <- sigma * sqrt(t)
36   c1 <- 2 * (r_d - r_f) / (sigma ** 2) + 1
37   c3 <- c1
38   c2 <- c1 + 1
```

```

32 -5:5 %>% apply(function(n){
33   if (callput == -1) {
34     d1n <- (log(spot * upper ** (2 * n) / (lower ** (2 * n + 1)))
35             + r_prime) / sigma_scaled
36     d2n <- (log(spot * upper ** (2 * n) / (strike * lower ** (2 * n))
37             + r_prime) / sigma_scaled
38     d3n <- (log(lower ** (2 * n + 2) / (lower * spot * upper ** (2
39             * n))) + r_prime) / sigma_scaled
40     d4n <- (log(lower ** (2 * n + 2) / (strike * spot * upper **
41             (2 * n))) + r_prime) / sigma_scaled
42   }
43   if (callput == 1) {
44     d1n <- (log(spot * upper ** (2 * n) / (strike * lower ** (2 * n))
45             + r_prime) / sigma_scaled
46     d2n <- (log(spot * upper ** (2 * n - 1) / (lower ** (2 * n)))
47             + r_prime) / sigma_scaled
48     d3n <- (log(lower ** (2 * n + 2) / (strike * spot * upper **
49             (2 * n))) + r_prime) / sigma_scaled
50     d4n <- (log(lower ** (2 * n + 2) / (spot * upper ** (2 * n +
51             1))) + r_prime) / sigma_scaled
52   }
53   c2n <- 2 * n / sigma ** 2
54
55   sum1 <- (upper ** n / lower ** n) ** c1 * (pnorm(d1n) - pnorm(d2n))
56   -
57   ((lower ** (n + 1)) / (upper ** n * spot)) ** c3 * (pnorm(d3n) -
58   pnorm(d4n))
59   sum2 <- (upper ** n / lower ** n) ** (c1 - 2) * (pnorm(d1n - sigma_
60   scaled) - pnorm(d2n - sigma_scaled)) -
61   ((lower ** (n + 1)) / (upper ** n * spot)) ** (c3 - 2) * (pnorm(
62   d3n - sigma_scaled) - pnorm(d4n - sigma_scaled))
63
64   c(sum1, sum2)
65 }) %>% rowSums %>% (function(sums) {
66   (callput * c(spot * exp(-r_f * t), -strike * exp(-r_d * t))) %*%
67   sums
68 })
69 }
70
71 dko_call <- function(spot, strike, upper, lower, r_d, r_f, sigma, t) {
72   dko(spot, strike, upper, lower, r_d, r_f, sigma, t, 1)
73 }
74
75 dko_put <- function(spot, strike, upper, lower, r_d, r_f, sigma, t) {
76   dko(spot, strike, upper, lower, r_d, r_f, sigma, t, -1)
77 }
78
79 #####
80 #dko(1.1, 1.10522, 1.155, 1.045, 0.5E-2, -0.2E-2, 11.08E-2, 0.5, -1)
81
82 package <- function(spot = 1.1, strike = 1.12, upper=1.21, lower=1.02,
83   leverage=1, maturity=0.5) {
84   r_d <- 0.5E-2

```

```

72 r_f <- -0.2E-2
73 sigma <- 11.08E-2
74 result <- leverage * dko_call(spot, strike, upper, lower, r_d, r_f,
75   sigma, maturity) +
76   leverage * dko_put(spot, strike, upper, lower, r_d, r_f,
77     sigma, maturity) +
78   plain_vanilla_call(spot, strike, r_d, r_f, sigma, maturity
79 ) -
80   plain_vanilla_put(spot, strike, r_d, r_f, sigma, maturity)
81 result
82 }
83
84 dko_package <- function(spot=1.1, sigma=11.08E-2, strike=1.13351, upper
85   =1.21, lower=0.99) {
86   r_d <- 0.5E-2
87   r_f <- -0.2E-2
88   maturity <- 0.5
89   result <- 100 * dko_call(spot, strike, upper, lower, r_d, r_f, sigma,
90     maturity) +
91     100 * dko_put(spot, strike, upper, lower, r_d, r_f, sigma, maturity)
92   result
93 }
94
95 #dko_package2 <- function(spot=1.1, sigma=11.08E-2, strike=1.13351, dist
96   =0.05) {
97   # r_d <- 0.5E-2
98   # r_f <- -0.2E-2
99   # maturity <- 0.5
100   # forward <- 1.10395
101   # worst_case <- strike
102   # upper <- worst_case + dist
103   # lower <- worst_case - dist
104   # result <- 100 * dko_call(spot, strike, upper, lower, r_d, r_f, sigma,
105     maturity) +
106     100 * dko_put(spot, strike, upper, lower, r_d, r_f, sigma, maturity)
107   # +
108   # plain_vanilla_call(1.1, strike, r_d, r_f, sigma, maturity) -
109   # plain_vanilla_put(1.1, strike, r_d, r_f, sigma, maturity)
110   # result
111   #}
112
113 ##### Zero Cost
114 #dko_package2(strike = 1.12, dist = 0.04531) #0 EUR
115 #dko_package2(strike = 1.12, dist = 0.04535) #2k EUR
116 #####
117
118 spot <- seq(1,1.25,0.005)
119 vol <- 11.02E-2
120 spot %>% sapply(function(x)dko_package(x,vol)) %>% plot(spot, y = .,
121   type = "l", main = "Value")
122 spot %>% sapply(function(x)dko_package(x,vol)) %>% diff %>% plot(spot
123   [-1], y = ., type = "l", main = "Delta")
124
125 vol <- seq(0.5E-2,30E-2,0.005)

```

```

116 vol %>% sapply(function(x) dko_package(sigma = x)) %>% plot(vol, y = .,
117 type = "l", main = "Value")
117 vol %>% sapply(function(x) dko_package(sigma = x)) %>% diff %>% plot(vol
118 [-1], y = ., type = "l", main = "Vega")
118
119
120 #####
121
122 spot <- seq(0.9, 1.35, 0.005)
123 spot %>% sapply(function(x) package(spot = x, upper = 1.2190, lower =
124 1.0230)) %>% data.frame(., spot)
125 plot(spot, y = ., type = "l", main = "Value")
126
127 writeOut <- function(name, fn) {
128   spot %>% sapply(fn) %>% multiply_by(1E7) %>% data.frame(spot, value=.)
129   %>% write.table(file = name, na = "nan", row.names = F, quote = F)
130 }
131
132 writeOut("unlev", function(x) package(spot = x, upper = 1.2190, lower =
133 1.0230))
134 writeOut("unlev_margin", function(x) package(spot = x, upper = 1.2190,
135 lower = 1.0243))
136 writeOut("lev", function(x) package(spot = x, upper = 1.1792, lower =
137 1.0608, leverage = 10))
138 writeOut("lev_margin", function(x) package(spot = x, upper = 1.1789,
139 lower = 1.0611, leverage = 10))
140
141 ## ——— OpenGL 3D
142 library(rgl)
143 spot <- seq(0.95, 1.25, 0.005)
144 vol <- seq(0.5E-2, 15E-2, 0.005)
145 l <- list(spot, vol)
146 comb <- do.call(expand.grid, l)
147 colnames(comb) <- c("Spot", "Vol")
148 head(comb)
149 value <- mapply(function(Spot, Vol) dko_package(Spot, Vol, strike=1.12,
150 upper=1.2190, lower=1.0230), comb$Spot, comb$Vol)
151
152 result <- cbind(comb, value)
153 result %>% write.table(file = "3dplot", na = "nan", row.names = F, quote
154 = F)
155 #wireframe(value ~ Spot * Vol, result)
156 persp3d(spot, vol, result$value, col = "skyblue", xlab = "Spot", ylab="
157 Volatility", zlab="Value")
158 ## ———
159
160 time <- seq(1/12, 1, 1/12)
161
162 c <- do.call(expand.grid, list(spot, time))
163 colnames(c) <- c("Spot", "Time")
164
165 value_delta <- mapply(function(Spot, Time) dko_package(Spot, Time,
166 strike=1.12, upper = 1.2190, lower = 1.0230), c$Spot, c$Time)
167 result_delta <- cbind(c, value_delta)

```

```

158
159 idx <- result_delta %>% group_by(Time) %>% attr(., "indices")
160
161 deltas <- idx %>% lapply(function(indices){
162   result_delta[indices,] %>% (function(x) diff(x$value_delta))
163 })
164
165 c_merged <- do.call(expand.grid, list(spot[-1], time))
166 colnames(c_merged) <- c("Spot", "Time")
167 final_result_delta <- cbind(c_merged, delta=unlist(deltas))
168 final_result_delta %>% write.table(file = "delta_surface", na = "nan",
169   row.names = F, quote = F)
169 persp3d(spot[-1], time, final_result_delta$delta, col = "skyblue", xlab =
  "Spot", ylab="Volatility", zlab="Value")

```