Pricing of European Convertible Bonds

111352027 Yu-Chen, Den

January 10, 2023

1 Brief introduction of Convertible Bonds

Convertible Bond

A convertible bond[1] is a fixed-income corporate debt security that yields interest payments, but can be converted into a predetermined number of common stock or equity shares. The conversion from the bond to stock can be done at certain times during the bond's life and is usually at the discretion of the bondholder.

Convertion Ratio

The conversion ratio[2] is the number of common shares received at the time of conversion for each convertible bond. The higher the ratio, the higher the number of common shares exchanged per convertible bond. The conversion ratio is determined at the time the convertible bond is issued and has an impact on the relative price of the security. The ratio is calculated by dividing the convertible bond's par value by the conversion price of equity.

The Formula for the conversion ratio is:

$$\text{Conversion Ratio} = \frac{\text{Par Value of Convertible Bond}}{\text{Conversion Price of Equity}}$$

2 Pricing

We can know that a convertible bond is construct by two different commodities

[1]Introduction to CB: https://www.investopedia.com/terms/c/convertiblebond.asp

[2]Introduction to CR: https://www.investopedia.com/terms/c/conversionratio.asp

- A fixed income Bond
- An European Call option

So we need to consider there two parts when we price a convertible bond. Where the price trends of fixed income bond is based of Vasicek model and the price trends of European Call option is based of Black-Scholes model.

Stock dynamic Process

Due to completeness of market, there is an equivalent martingale measure Q under which the average interest rate is equals to risk-neutral interest rate r_t. Therefore, the changing stcok process is

$$dS_t = r_t S_t dt + \sigma_{s,t} S_t dW^Q_{s,t}$$

$$S_T = S_t e^{(r-\frac{1}{2}\sigma^2_{s,t})(T-t)+\sigma_{s,t} dW^Q_{s,T-t}}$$

Where

- S_t is the stock price at time t
- r is the risk-neutral interest rate under Q measure
- $\sigma_{s,t}$ is the volatility of stock return
- $dW^Q_{s,T-t}$ is the Brownian motion under Q measure $\sim N(0, d(T-t))$

Bond dynamic Process

Assume that the return of bond price P_t follows the below process

$$\frac{dP_t}{P_t} = r_t dt + \sigma_{p,t} dW^Q_{p,t}$$

Where

- P_t is the bond price at time t
- r_t is the risk-neutral interest rate under Q measure
- $\sigma_{p,t}$ us the volatility of bond return

- $dW^{Q}_{p,T-t}$ is the Brwonian motion under Q measure $\sim N(0, d(T-t))$

$$P_T = P_t e^{(r_t - \frac{1}{2}\sigma^2_{p,t})(T-t) + \sigma_{p,t} dW^{Q}_{p,T-t}}$$

Value of Convertible Bond

The value of convertible bond is separate into 3 conditions[1]

$$V(T, S_T) = \begin{cases} P_b & \text{if } S_T \leq C_v \\ P_b & \text{if } C_v < S_T \leq \frac{P_b}{M} C_v \\ \frac{M}{C_v} S_T & \text{if } S_T > \frac{P_b}{M} C_v \end{cases}$$

Where

- P_b is the price of the bond
- C_v is the convertible price of stock
- M is the face value of the bond
- S_T is the market stock price at time T

We consider about the convertible premium rate[3] $\frac{P_b}{M}$, which the price of a convertible security exceeds the current market value of the common stock into which it may be converted. A conversion premium is expressed as a dollar amount and represents the difference between the price of the convertible and the greater of the conversion or straight bond value.

3 Data

Data description

We use 23383 (3rd domestic convertible bond of 2338.TW) which established in 2021/08/03, initial convert date is 2021/11/04, and the maturity date is 2026/08/03 as our simulate target. The underlying stock data is from api of yahoo finance **yfinance**[4], and the bond

[3] definition: https://smart.businessweekly.com.tw/Reading/WebArticle.aspx?id=7006608&p=2

[4] https://pypi.org/project/yfinance/

data is from **MOPS**[5]. We will use some parameters to simulate the price of convertible bond, which are

- Face Value: $100,000$
- Issue Price: $115,230$
- conversion price: 88.8
- convertible premium rate: 1.0242
- risk free rate: 1.5%

4 Empirical Result

We can price the convertible bond by using the following steps

- Calculate the bond price P discounted by risk free rate
- Calculate the call price C by bond's market value minus the initial bond value
- Calculate the convertible premium rate C_r by $\frac{C+P}{M}$
- Simulate the stock price S by using historical volatility
- After the initial convert date, we can get the convertible bond value V by comparing the stock price and conversion value.

Then we can get the simulation price of stock by Monte-Carlo simulation with $N = 1000$

[5] https://mops.twse.com.tw/mops/web/t120sb02_q9

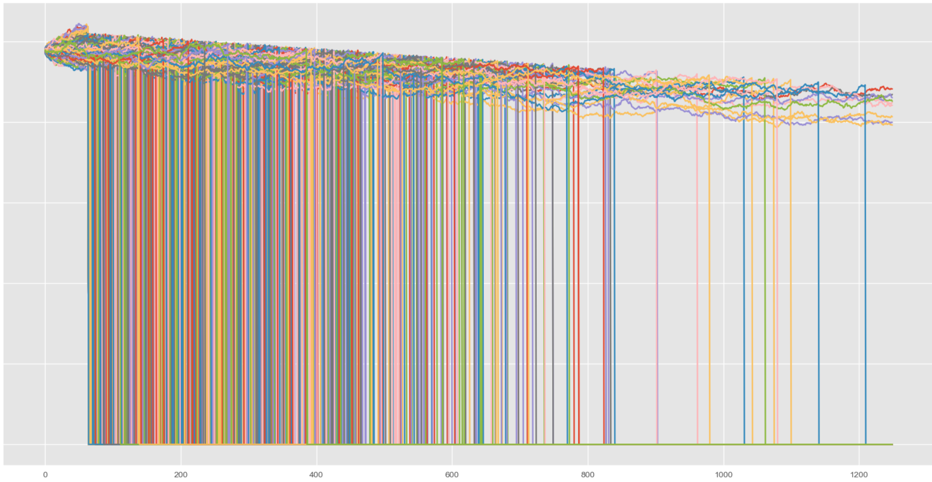

Figure 1: Simulation price of stock

When the stock price drops to 0, it means that the CB convert to stock and the contract is terminated. In this project, we simulate 1000 times of CB's value and get the expected initial value of 92,892.24, and the true initial value of CB is 115,230, the primium rate is 19.39%.

5 Conclusion

This is the most simple method to price the CB, in some papers, they use Least Square Monte-Carlo simulation to price the CB beacuse it has the characteristic of early exercise, just like the graph above, and some also use the trinomial tree method and assume that the interest rate r follows the Vasicek model. So, my future work will focus on the interest rate process and the early exercise characteristic of CB.

References

[1] Jingyi Shao and Wanyi Chen. Empirical analysis of martingale pricing for convertible bonds based on vasicek model. In *2019 Chinese Control And Decision Conference (CCDC)*, pages 4173–4178. IEEE, 2019.

Appendix

```python
import polars as pl
import numpy as np
import yfinance as yf
import matplotlib.pyplot as plt
import os
import sys
from numpy import exp as e
from numpy import sqrt as sqrt

stock = yf.download('2338.TW', start = '2020-08-03', end = '2021-08-03')
close = stock['Close']

plt.style.use('seaborn')
plt.plot(close, label = '2338.TW')
plt.legend()
plt.show()

'''define parameters'''
S = np.zeros([1000, 1250])
P = np.zeros(1250)
P0 = 115230
C_v = np.array([88.8] * 1250)
M = 100000
r = 0.015
sigma = close.pct_change().std()
dt = 1/250
V = []
C = np.zeros(1250)

'''Bond price'''
for t in range(1, 1251):
    P[-t] = M * e(-r * dt * (t-1))
```

```
'''call price'''
C0 = P0 - P[0]
de = C0 / len(P)
for t in range(1250):
    C[t] = C0 - de * (t + 1)

'''Monte Carlo Simulation'''
C_r = (C + P)/M
for N in range(1000):
    S_0 = close[-1] ## S_0 is the first value of close price

    for t in range(1250):
        dW_t = np.random.normal(0, np.sqrt(dt))
        S[N, t] = S_0 * e((r - 0.5 * sigma ** 2) * dt + sigma * dW_t)

        if S[N, t] > C_v[t] * C_r[t] and t > 0.25*250:
            V.append(((S[N, t] - C_v[t] * C_r[t]) * e(-r*dt*t)) + P[0])
            break
        elif t == 1249:
            V.append(P[0] + C[0])
        else:
            S_0 = S[N, t]

'''fig1'''
plt.style.use('ggplot')
plt.figure(figsize = (20, 10))
for i in range(1250):
    plt.plot(S[i])
plt.xticks(np.arange(0, 1250, 50), np.arange(0, 5, 0.2))
plt.show()

'''V'''
print(np.mean(V))
```

```
premium_ratio = (P0 - np.mean(V))/P0 * 100
print(f'premium ratio: {premium_ratio:.2f}%')
```