May 22, 2016

Global Credit Products - Case Study Report

Mufan Li
mufan.li@mail.utoronto.ca

1 Convertible Bond Pricing

This section describes the solution to the first part of the case study on pricing of a convertible bond. Please refer to the attached OTPP_Convertible_Bond.xlsm file for the computation in Excel VBA. The specific details will be explain in Section 1.3.

1.1 Description of a Convertible Bond

A *convertible bond* is a derivative security that depends on the underlying stock price. Specifically, the convertible bond behaves as a regular bond while also providing the holder a right to convert the bond into stock at a prespecified ratio. For this problem, our convertible bond will continue to pay a semiannual coupon until maturity without exercise of the conversion (and default); when exercised, each bond will convert into a number (*conversion ratio*) of stock.

Since exercise is allowed at anytime, the security behaves like an American option: it will always take the maximum of the holding value and the exercise value (also known as *parity price*). Due to this complexity, there does not exist a closed-form solution for the price, and we must rely on a numerical approximation instead.

It is also common for the convertible bond to be callable or puttable; a *callable bond* allows the issuer of the bond to buy back the bonds before the maturity date; similarly a *puttable bond* allows the holder of the bond to demand early repayment of bond. In this problem, we will not deal with the complexity of such instruments.

1.2 The Binomial Tree Approach to Pricing

A *binomial tree* (also known as *binomial lattice*) is a discretized model of a continuous stochastic process, such that at any stage, the process can only evolve to two potential values in the next step. As the size of the time steps is taken to zero, the discretized process will converge to the continuous process (under some technical conditions). In practice, the convergence is more than sufficient for modeling simple derivative prices.

In this problem, we allow a third potential value in the event of a default, where the stock price will become zero, and the bond price will be equal to the *recovery value*. The underlying principle remains the same for find the price of the convertible bond P_{cb}

$$P_{\text{cb}}(t, S(t)) = \max \left\{ P_{\text{exercise}}(t, S(t)), P_{\text{hold}}(t, S(t)) + P_{\text{coupon}}(t) \right\} \tag{1}$$

where $P_{\text{coupon}}(t)$ is the coupon payment, and $P_{\text{exercise}}(t, S(t))$ is the value of the conversion at conversion rate C_r

$$P_{\text{exercise}}(t, S(t)) = C_r S(t)$$

$P_{\text{hold}}(t, S(t))$ can be computed as the discounted (conditional) expectation of the price at the next time step

$$\begin{aligned} P_{\text{hold}}(t, S(t)) &= e^{-r\Delta t}\mathbb{E}\Big[P_{\text{cb}}(t+\Delta t, S(t+\Delta t))\Big| S(t)\Big] \\ &= e^{-r\Delta t}\Big[p_{\text{up}}P_{\text{cb}}(t+\Delta t, S_{\text{up}}(t+\Delta t)) + p_{\text{down}}P_{\text{cb}}(t+\Delta t, S_{\text{down}}(t+\Delta t)) + p_{\text{default}}P_{\text{recovery}}\Big] \end{aligned} \tag{2}$$

Here $p_{\text{up}}, p_{\text{down}}, p_{\text{default}}$ are (risk-neutral) probabilities of each of the branches in the binomial tree, r is the risk-free rate, and $S_{\text{up}}, S_{\text{down}}$ correspond to the up and down branches of the binomial tree of the stock.

Observe that $P_{\text{cb}}(t, S(t))$ is can be found recursively from the values of $P_{\text{cb}}(t+\Delta t, S(t+\Delta t))$. Therefore, by determining the values of $P_{cb}(T)$ where T is the maturity time. Since the holding value of convertible bond at maturity is just the par value plus the final coupon, we have

$$P_{\text{cb}}(T, S(T)) = \max\Big\{C_r S(T), P_{\text{par}} + P_{\text{coupon}}(T)\Big\} \tag{3}$$

With this terminal condition, we can recursively find $P_{\text{cb}}(T-\Delta t), P_{\text{cb}}(T-2\Delta t), \ldots, P_{\text{cb}}(0)$, where the present value of the bond is found at $t = 0$.

1.3 VBA Implementation

In the VBA code, we will work with two arrays *ConvertNext* and *ConvertCurrent*, corresponding to the values of $P_{\text{cb}}(t+\Delta t)$ and $P_{\text{cb}}(t)$. And as we step backward in time, the two arrays will be resized as needed.

We start by applying the terminal condition in Equation (3), which is implemented by the following code

```
For State = 0 To N_steps
    stock = S0 * (u ^ State) * (d ^ (N_steps - State))

    ConvertNext(State) = Application.Max(conv_ratio * stock, _
                                         par + coupon_val)
Next State
```

Listing 1: VBA Code for Terminal Condition

After which, we can then apply the conditional expectation in Equation (2) to assign the value of variable *Convert*. Finally *ConvertCurrent* is determined by applying the exercise constraint in Equation 1.

```
For State = 0 To Index
    stock = S0 * (u ^ State) * (d ^ (Index - State))
```

```
    Convert = Pd * ConvertNext(State) + Pu * ConvertNext(State + 1) _
              + (1 - Exp(-lambda * dt)) * Recovery * par

    ConvertCurrent(State) = Application.Max(conv_ratio * stock, _
                            Exp(-R * dt) * Convert + coupon)
Next State
```

Listing 2: VBA Code for Terminal Condition

To finish the iteration, we copy the values in *ConvertCurrent* to *ConvertNext* before the next loop.

After exiting the loop, we will finally end up with only one node in the binomial tree, which is the value of *ConvertCurrent(1,1)* corresponding to $P_{\text{cb}}(0)$.

1.4 Computing Delta

The *delta* of the convertible bond is

$$\Delta = \left.\frac{\partial P_{\text{cb}}(t, S(t))}{\partial S(t)}\right|_{t=0}$$

For this part of the problem, we consider two approaches. First by centered finite difference approximation, we know that

$$\frac{\partial P_{\text{cb}}(t, S(t))}{\partial S(t)} = \frac{P_{\text{cb}}(t, S(t)+\epsilon) - P_{\text{cb}}(t, S(t)-\epsilon)}{2\epsilon} + \mathcal{O}(\epsilon^2) \tag{4}$$

where $\mathcal{O}(\cdot)$ refers to the order of the discretization error. With this method, we have a convergence guarantee by Taylor's Theorem; however, this requires two additional computations of the binomial trees.

Since the binomial tree is already computed once, we can then use the first branches to approximate the delta as well to save computations. Here we use the following approximation

$$\frac{\partial P_{\text{cb}}(t, S(t))}{\partial S(t)} = \frac{P_{\text{cb}}(t+\Delta t, uS(t)) - P_{\text{cb}}(t+\Delta t, dS(t)-\epsilon)}{(u-d)S(t)} + \mathcal{O}\left(\Delta t + [(u-d)S(t)]^2\right) \tag{5}$$

where u, d are the returns of the up and down branches.

There are two issues with this approximation. Firstly the error is capped by the discretization size in time instead of a user chosen ϵ. Although this can be resolved by choosing a very small Δt in the beginning to improve the approximation, it will require a different implementation. Secondly, this approximation ignores the third branch in the event of a default. Therefore, we expect this estimate to be worse while not repeating computation.

When comparing the two estimates for this problem, the difference is quite small:

Finite Difference: 1.8432617

Binomial Branches: 1.8411774

where we chose $\epsilon = 10^{-6}$. Therefore we can conclude that for this problem, it is sufficient to approximate by using the first branches in the binomial tree.

2 Trend Following Model

This part of the report investigates the trend following models. The script file that drives combines all the functions is in *trend.m*, while all the other *.m* defines the helper functions as required. Note here *helper.m* implements a placeholder class such that all the extra helper functions can be combined in one *.m* file.

Model Assumptions (Interpretation)

- When computing the CDS spread for 5 year maturity, it is assumed that the CDS spread desired is the CDS contract with maturity of exactly 5 years. This value is linearly interpolated between the 5 year standard contract (which always has a maturity of longer than 5 years), and 4 year standard contract (which has a maturity of less than 5 years).
- When computing the returns of each CDS series, it is assumed that *cv01(s)* can be *extrapolated* when the spread is greater than 5000, as the CDS series 59 does at one point reach a spread of greater than 10000.
- When implementing the trading rule, it is assumed that "sell protection 1 million CDS notional" refers to taking a short position (long risk) in the CDS, instead of adding an additional 1 million short position into the portfolio. Otherwise, this will potentially lead to trading every CDS on every day.

2.1 The Basic Model and Summary Statistics

Here we look at several summary statistics of the basic strategy. Return is calculated as profit in dollars as a percent of notional value ($ 1 million), assuming we do not re-invest the profit, which results in the cumulative profit computed as a sum of dollar profits instead of a cumulative product of returns. This assumption is used since returns in CDS are in the form of a cash flow, however it will have minimal influence on the analysis. The Sharpe ratio calculation is computed based on daily returns annualized and using zero risk-free rate. The kurtosis refers to the non-excess kurtosis, and both skewness kurtosis are computed for daily portfolio returns.

Statistic	Value
Cumulative Return	7.01%
Annualized Return	2.87%
Sharpe Ratio	4.0256
Skewness	-0.2167
Kurtosis	14.4875

Table 1: Summary Statistics of the basic strategy.

Based on these statistics alone, especially the Sharpe ratio, this strategy does look attractive. However, the kurtosis is relatively large even when compared to a t-distribution with 5 degrees of freedom (kurtosis of 9), indicating a very heavy tail.

2.2 Number of Portfolio Components

The most basic model in the previous part used all 343 CDS series to compute the return, when it is potentially unnecessary. While clearly diversifying can reduce volatility of the portfolio, which improves the Sharpe ratio, there should be a point at which the improvement will begin diminishing. We can investigate the performance of portfolio of different sizes and seek an approximate optimal size. Since we do not have external information to select the best CDS series, we can look at random samples from the 343 series and compute their return statistics.

Specifically we randomly choose N_{sample} out of $N = 343$, a total of 50 times, and compute the summary statistics for all 50 scenarios. We can use the box plot to observe the change in distributions for the annualized return and Sharpe ratio as we vary the value of N_{sample}.

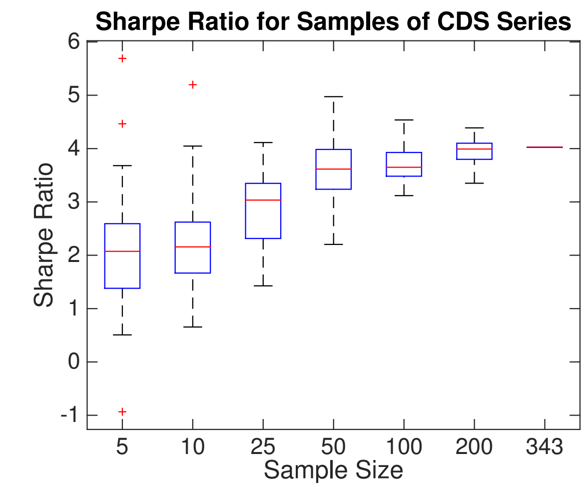

Figure 1: The box plot of Sharpe ratio for sampled portfolio over different sample sizes.

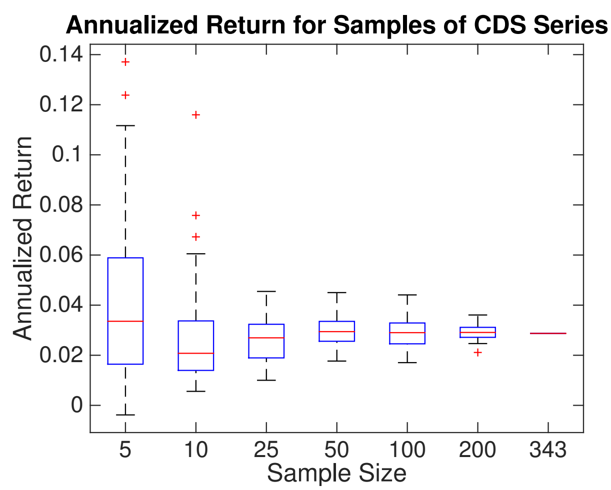

Figure 2: The box plot of annualized return for sampled portfolio over different sample sizes.

From the box plots, we can observe the improvement in average Sharpe ratio begins to diminish after reaching a sample size of 50. Therefore, we can conclude that it is unnecessary to trade all 343 CDS series, instead 50-100 components will be able to reach similar results.

2.3 Missing Assumptions and Transaction Cost

A key difference between back-testing and actual trading is the execution cost due to available liquidity, trade size, and broker commission. Since there is no additional information on individual CDS series and date, it is appropriate to assume a uniform transaction cost on each trade.

Here we assume the transaction cost is directly proportional to the notional value of the trade, which implies that each time a portfolio position flips from long $1 million to short $1 million, we have traded $2 million of notional value. After several guesses, it is easily found that the break-even transaction cost is approximately 5.41 basis points. We can look at the summary statistics again:

Statistic	Value
Cumulative Return	0.00%
Annualized Return	0.00%
Sharpe Ratio	-0.0010
Skewness	-0.1464
Kurtosis	14.1028

Table 2: Summary Statistics of the basic strategy with transaction cost of 5.41 basis points.

Clearly, a transaction cost of 5.41 basis points is very small and unrealistic. Therefore the strategy, in its current state, is not very attractive.

2.4 Strategy's Potential

If 5.41 basis points of transaction cost is enough to break even an annualized returns of 2.87%, clearly direct application of the strategy induces a high trading volume. Perhaps some of the trading activity is unnecessary. We can get a rough idea of trading activity by simply looking at the sparsity of the matrix of trading volume.

Here we observe Figure 3 that the strategy tend to frequently reverse its position on a single CDS series, as often as 4 days in a row. This type of trading is due to the moving average oscillating between positive and negative values, causing the strategy to trade frequently. Therefore, the strategy has potential to significantly reduce its transaction costs by eliminating this type of trades.

Additionally, the choice of using a 10-day moving average is arbitrary. This length of moving average can be optimized for better results.

2.5 Optimized Strategy

As mentioned in the previous section, the strategy can be improved by both reducing unnecessary trading volume, and by optimizing the length of simple moving average.

Here we investigate a simple modification to the trading rule by introducing a tolerance to the moving average, where we will continue to hold the current position until the moving average has

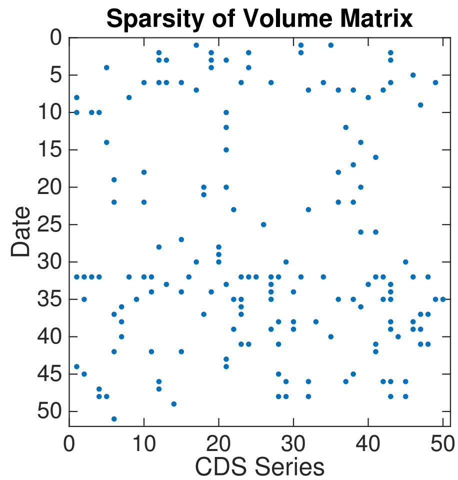

Figure 3: A (sample) sparsity plot of the trading volume matrix, where each point indicates a non-zero entry in the matrix, i.e. a trade.

crossed to the opposite sign by at least some $\epsilon_{\text{tol}} > 0$. More formally

$$\text{Position on CDS}_i = \begin{cases} -1 & \text{if } \text{MA}_t(S_i, T) > \epsilon_{\text{tol}} \\ +1 & \text{if } \text{MA}_t(S_i, T) < -\epsilon_{\text{tol}} \\ \text{Hold previous position} & \text{otherwise} \end{cases}$$

where T is the length of the moving average.

Here we can formulate the strategy choice as an optimization problem over the parameters $\{T, \epsilon_{\text{tol}}\}$, with the objective to maximize some combination of returns and Sharpe ratio. Since the choice of T has very little impact over small changes, and minimization over one variable is a much simpler problem, we decide to pursue the following formulation:

$$\epsilon^*_{\text{tol}}(T) = \underset{\epsilon_{\text{tol}} \in [0, 0.01]}{\arg\max}\ 100 r_{\text{annual}}(T, \epsilon_{\text{tol}}) + R_{\text{Sharpe}}(T, \epsilon_{\text{tol}}) \tag{6}$$

where the optimization problem will be solved over several choices of T. Here r_{annual} is the annual return, R_{Sharpe} is the annualized Sharpe ratio, and the factor 100 is chosen so the two values are on the same order of magnitude.

MA Length (T)	3	5	7	10	15	20	30	50
Tolerance (ϵ_{tol})	0.0024	0.0040	0.0061	0.0043	0.0068	0.0062	0.0034	0.0038
Cumulative Return	1.47%	1.14%	0.19%	0.54%	-0.01%	0.06%	-0.14%	-0.16%
Annualized Return	**1.21%**	0.93%	0.15%	0.44%	-0.01%	0.05%	-0.12%	-0.13%
Sharpe Ratio	2.1592	**2.3534**	0.5359	1.3905	-0.0378	0.2265	-0.4462	-0.7317
Skewness	0.4409	1.2342	0.4260	0.7464	-0.3771	-0.2468	-0.4187	-0.8695
Kurtosis	7.1357	9.4457	6.6211	6.8286	7.3626	8.0430	5.8006	7.4253

Table 3: Out-of-sample summary statistics of the optimized strategy with transaction cost of 20 basis points, and over several different choices of moving average length.

The implementation of this optimization is done through MATLAB function *fminbnd*, assuming a transaction cost of 20 basis points. Here we also separate the data into in-sample for optimization, and out-of-sample for testing. In this experiment, we choose the first half of CDS series (1-172), and the first half of the time period as in-sample, and the second half of CDS series (173-343) and the second half of the time period as out-of-sample. This choice results in no overlap of both CDS series and time period between the in-sample and out-of-sample data. The code for this section can be found in *improveStrategy.m*.

Observe the optimization and summary results in Table 3, we see that even with a transaction cost of 20 basis points, we can still reach a Sharpe ratio of 2.35 with an optimized strategy. Interestingly, we also find the strategy improved in terms of kurtosis as well. Similarly, we can observe the sparsity plot of the volume matrix with 3-day moving average. Even with a choice of small T, we can have restrict the trading activity to less than previously in Figure 3.

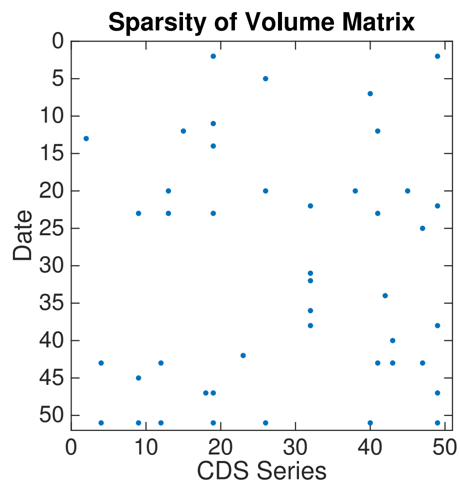

Figure 4: A (sample) sparsity plot of the trading volume matrix for the optimized strategy with 3-day moving average, where each point indicates a non-zero entry in the matrix, i.e. a trade.

In conclusion, we find the optimized strategy can perform even under realistic transaction cost level at 20 basis points. The strategy at this stage is definitely worth pursuing.