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