

MAFS5250 – Computational Methods for Pricing Structured Products

Computer Assignment One

Instructor: Prof Y.K. Kwok

* Work in a group of two.

Pricing Behavior of an Equity-Linked Structured Product

You are invited to explore the pricing behavior of an equity-linked structured product

“24-month callable dual accrual cash or share security on
Wal-Mart Stores, Inc and Intel Corp”

launched by Merrill Lynch in 2008. The product description is outlined below:

Issue size:	10,000,000 warrants
Minimum subscription:	100,000 warrants
Notional Amount:	USD 1 per warrant
Issue Price:	100% of the Notional Amount
Valuation Date:	Feb. 11, 2006
Maturity Date:	Feb. 19, 2008

The two underlying stocks are

	<i>Reference price</i>	<i>Exercise price</i>
Wal-Mart Stores Inc.	USD 45.48	USD 39.5676
Intel Corp	USD 20.77	USD 18.0699

where Exercise Price = 87% x Reference Price. The Reference Price is taken to be the closing price of the stock on the valuation date. Note that the two stocks have been chosen such that the price processes of them are expected to exhibit minimal correlation.

1. Payoff structure:

Full par payment or delivery of the “worst performing” stock on the maturity date.

- If the settlement prices of BOTH the underlying stocks are higher than or equal to the respective Exercise Price, then each warrant holder receives 100% of the notional amount per warrant held.
- If either one of the settlement prices is lower than the respective exercise price, then each holder receives per warrant physical delivery of a number of the “Worst performing” stock equal to
$$\frac{\text{Notional amount}}{\text{Exercise Price}}$$
of the worse performing stock.

$$\begin{aligned}
& \text{Terminal payoff at time } T \\
& = \min(1, \min(S_1(T) / S_{1, \text{exer}}, S_2(T) / S_{2, \text{exer}})) \\
& = 1 - \max(1 - \min(S_1(T) / S_{1, \text{exer}}, S_2(T) / S_{2, \text{exer}}), 0),
\end{aligned}$$

where $S_{1, \text{exer}}$ and $S_{2, \text{exer}}$ are the Exercise Price of Stock 1 and Stock 2, respectively. That is, the investor shorts a put on the minimum of the two stocks.

This is a contingent forced conversion which occurs when either one of the two share prices declines. This is just the **opposite** to that of a convertible bond where the holder of a convertible bond chooses to convert the bond par into shares only when the share price appreciates above certain threshold value.

*Query: Would the **chance** of occurrence of contingent forced conversion become higher or otherwise when the **correlation** between the price processes of the two stocks becomes closer to zero?*

2. Additional coupon (accrual feature)

The warrant pays out a fixed 4.075% **coupon** for the first quarter (that is, 16.3% per annum). Afterwards, unless the warrant has been called, over each observation period (3-month period), the holder receives

$$4.075\% \times n / N \text{ of notional amount}$$

where

N = number of New York Business Days in the period in the applicable Observation Period;

n = number of New York Business Days in the applicable Observation Period on which the closing prices of BOTH the Underlying Stocks are **at or above** the respective Exercise Price.

This is like an accrual note with the underlying index being the minimum of the two share prices. The accrual feature can be viewed as a series of daily **binary** options, and the warrant pays at the n th time step

$$4.075\% / N \times \text{notional amount}$$

when

$$\min(S_1(n, i) / S_{1, \text{exer}}, S_2(n, j) / S_{2, \text{exer}}) > 1,$$

where $S_1(n, i)$ is the price of Stock 1 at n time steps from initiation and i up moves, and $S_2(n, j)$ is the price of Stock 2 at n time steps from initiation and j up moves.

Remark

The accrual feature is path dependent since the number of days that coupon will be received depends on the realization of the stock price processes (both have to stay above their respective Exercise Price). It is straightforward to use the **forward shooting grid** technique to perform the day counting (in a similar manner to that

of the Parisian feature). The corresponding grid function at each lattice node can be constructed as follows

$$g_{coupon}(n, i, j, k) = k + \mathbf{1}\{\min(S_1(n, i) / S_{1, exer}, S_2(n, j) / S_{2, exer}) > 1\}.$$

Here, k counts the number of days over each 3-month Observation Period, when the condition

$$\min(S_1(n, i) / S_{1, exer}, S_2(n, j) / S_{2, exer}) > 1$$

is satisfied. The index k is reset to zero immediately after a coupon date where the accrual coupons have been added to the warrant value.

3. Issuer's Call:

On any of the Observation Date over each 3-month Observation Period, provided that **BOTH** underlying stocks are greater than or equal to the reference prices, the issuer can call by paying **100%** of the **Notional** Amount (together with the **coupons accrued** in the last 3-month Observation period). The Observation Dates are set to be at the end of each 3-month period over the life of the warrant. For convenience, we assume that the Coupon Dates and the Observation Dates coincide. Similar to an American option, it is **NOT always optimal** for the issuer to call when both stock prices are above the reference prices (currently in-the-money). Indeed, the optimal calling policy adopted by the issuer is determined as part of the solution procedure.

To incorporate the call feature, at each lattice node on an Observation Date, we apply the dynamic programming procedure: $\min(W_{cont}, K)$, where W_{cont} is the continuation value of the warrant and K is the call price (taken to be 100% of the Notional Amount).

Comments on the nature of the product

warrant = bond (series of binary options due to the accrual feature of coupons)

- European put on minimum of two uncorrelated stocks
- issuer's calling right (Bermudan call option with multiple call dates)

- The investor **believes** that the prices of BOTH underlying shares at maturity will remain at a level above or equal to their respective Exercise Prices, earning an enhanced yield.
- Note that $\min(W_{cont}, K) = W_{cont} - \max(W_{cont} - K, 0)$. This “call” right given to the issuer is like a Bermudan call option with strike price equal to the call price. The call price is 100% of the Notional.
- The coupons received depend on the trading path of BOTH underlying stocks due to the accrual feature. The coupons are calculated based on a series of binary option payoffs.

Sources of risks faced by the investor

1. Market risks – stochastic movement of the prices of the underlying shares

2. Interest rate risk – the present value of the bond component, including par plus coupons.
3. Issuer's call.
4. Counterparty risk – default of Merrill Lynch (more noticeable after the event of Lehman Brothers' minibonds)
5. Liquidity risk – will not be listed on any securities exchange and do not expect a trading market with only Merrill Lynch as a possible buyer.

Work elements in this computer assignment

1. Construct the two-state lattice tree algorithm for pricing this two-state option product (warrant), taking into consideration of (i) terminal payoff structure, (ii) accrual feature of the coupons, (iii) issuer's call right. Constant interest rate and zero default risk of the issuer are assumed. In your report, you are required to describe the special considerations that have been taken to incorporate the issuer's call, accrual coupons and terminal payoff structure in your scheme.
2. Examine the variation of the warrant's price with respect to the following parameters:
 - (i) correlation coefficient between the stock price processes,
 - (ii) volatility of the stock prices,
 - (iii) level of the riskless interest rate.
 Plot the warrant price against each of the above parameters. Give your comments on the pricing behavior of the warrant. Do the computed results coincide with your financial intuition?
3. Compute the value of the embedded issuer's call right by finding the difference of the warrant's price with and without the call right. How does the value of the call right change with varying model parameters?

Hints on the construction of the two-state trinomial scheme

1. We take the number of trading days in a year to be 252 so that one quarter of a year is 63 days. It suffices to take each time step to be 3 days so that the 2-year term of the warrant corresponds to total number of time steps equal to 168.
2. Let $W(n, i, j, k)$ denote the warrant value at n time steps from initiation, i up moves for Stock 1, j up moves for Stock 2 and k days that have been counted from the last coupon payment date up to the current day such that the accrual coupons will be collected on the next coupon payment date. The range of values assumed by the indexes are: $n = 1, 2, \dots, 168$; $i = -n, -n + 1, -1, 0, 1, \dots, n - 1, n$; $j = -n, -n + 1, -1, 0, 1, \dots, n - 1, n$; $k = 1, 2, \dots, K_n$, where K_n is the number of days lapsed from the last coupon payment date.
3. The two-state lattice tree algorithm is constructed as

$$W(n, i, j, k) = [p_{uu} W(n+1, i+1, j+1, g(n, i+1, j+1, k)) + p_{ud} W(n+1, i+1, j-1, g(n, i+1, j-1, k)) + p_{du} W(n+1, i-1, j+1, g(n, i-1, j+1, k))]$$

$$+ p_{dd} W(n+1, i-1, j-1, g(n, i-1, j-1, k)) \\ + p_{00} W(n+1, i, j, g(n, i, j, k))] / R, \quad \textcircled{?}$$

where p_{uu} denotes the probability that both Stock Prices move up, and a similar notation for other probability values, and $1 / R$ is the discount factor over one time period. These probability values have been documented in the lecture note.

4. When $n = 21, 42, \dots, 147$, the holder is entitled to receive the accrual coupon. Also, the issuer may call on these dates. Note that the coupon will always be received by the holder even upon calling by the issuer. That is, the total cash amount received by the holder upon calling is the notional plus the accrual coupon. The actual amount of the accrual coupon is dependent on the realization of the stock price processes during the Observation Period, which is the time interval between this coupon date and the last coupon date. As your main contribution to the formulation of the numerical scheme in this project, you are required to construct the corresponding lattice tree algorithm that takes special considerations to incorporate the dynamic programming procedure with regard to the issuer's call, jump in warrant value due to coupon payment, and resetting of the day counting of accrual coupon.
5. The maturity date corresponds to $n = 168$. The terminal payoff is the par minus the put on the minimum of the two stock prices, plus the last coupon accrued between $n = 148$ and $n = 168$. It is necessary to find the terminal payoff at all varying values of i, j and k . The range of each of these indexes shrinks as we perform the successive backward induction calculations. Note that k ranges from 1 to 21 over each successive coupon accrual period (one quarter of a year), and k drops to zero right after a coupon payment date. Also, the two indexes i and j can assume only the value zero at the tip of the binomial tree.
6. As there is no intermediate knock-out (barrier feature), the use of the two-dimensional trinomial feature is acceptable. If otherwise, special precautions are required to implement the boundary conditions along the 4 sides of the computational domain in order to incorporate the knock-out feature.
7. The time step has been chosen to be 3 days. You have the freedom to choose the stepwidth for the two state variables $\ln S_1$ and $\ln S_2$. Be careful that you choose the computational domain which spans sufficiently large positive and negative values for both state variables. Recall that the domain of definition of the continuous model is the whole infinite $(\ln S_1 - \ln S_2)$ plane. The error associated with the truncation of the domain can be quite significant if the span of the computational domain is not sufficient.

Remark

Without the issuer's call feature, there will be no necessity to use the forward shooting grid algorithm to keep track of the coupons accumulated. Apparently, a simpler

numerical procedure is to add the present value of the coupon accrued at the lattice node since the coupon is sure to be collected once the two asset prices satisfy the coupon payment condition. The discount factor used is based on the time interval between the date of accrual and the Observation Date (date of collection).

When there exists the *calling right* held by the issuer, the numerical algorithm needs to choose the minimum among the continuation value W_{cont} and call price K at each lattice node on a call date. The coupons may not be collected when the investor is forced to collect K when $W_{cont} > K$. This explains why the forward shooting grid method is required with the presence of calling right.

You are invited to check validity of the above argument by **comparing** the numerical contract values obtained using the following two approaches: (i) use of the forward shooting grid but without the issuer's call feature, (ii) add the present value of the coupon directly at each node provided the coupon accrual condition is met.