Valuation Project 1: Binomial Tree

Brian Wu / UID: bhwu3@illinois.edu / UIN: 664883556

Executive summary

Product: #4 Autocallable Phoenix Securities Based on the S&P 500® Index
Issuer: Citigroup Global Markets Holdings Inc. (CGMI)
Estimated value per security by CGMI: $987.80
Estimated value per security for Project #1: $988.71

The product is an auto-callable security with roughly one year to maturity and with four coupon payments on four designated dates (also auto-callable on these dates). Noted that the coupon payments have a path-dependent feature, therefore accurately pricing such product with binomial tree model seems feasible.

The binomial model of choice is Cox, Ross, and Rubinstein (CRR), since it provides a nice property and is fairly accurate in valuing securities as time step increases. For this project, using 3770 steps in CRR model yields a value of $988.71 for the valuation model.

Data collection and parameters used

Choosing an integer number, N, times 377 as time steps makes a good fit for the model because 377-day is the duration of the securities, making all the interim dates and final valuation date right on their respective time steps, which will free us from worrying about fitting non-integer time steps into the tree.

The risk-free rate is 0.03833 quoted from Bloomberg terminal's OIS rate. After converting to continuously compounded rate, we get 0.0381027 for the binomial tree model's risk-free rate.

The dividend yield is 0.01642 also from Bloomberg terminal.

As for sigma, 0.23441 is chosen from a pool of reasonable sigma from Bloomberg. For each of the four interim dates and final valuation date, get the volatilities for moneyness equals to 80 and 100, which make up to eight reasonable volatilities. After running a sensitivity test for each volatility, sigma 0.23441 yields a security value that is closest to the estimated value by CGMI.

Key terms

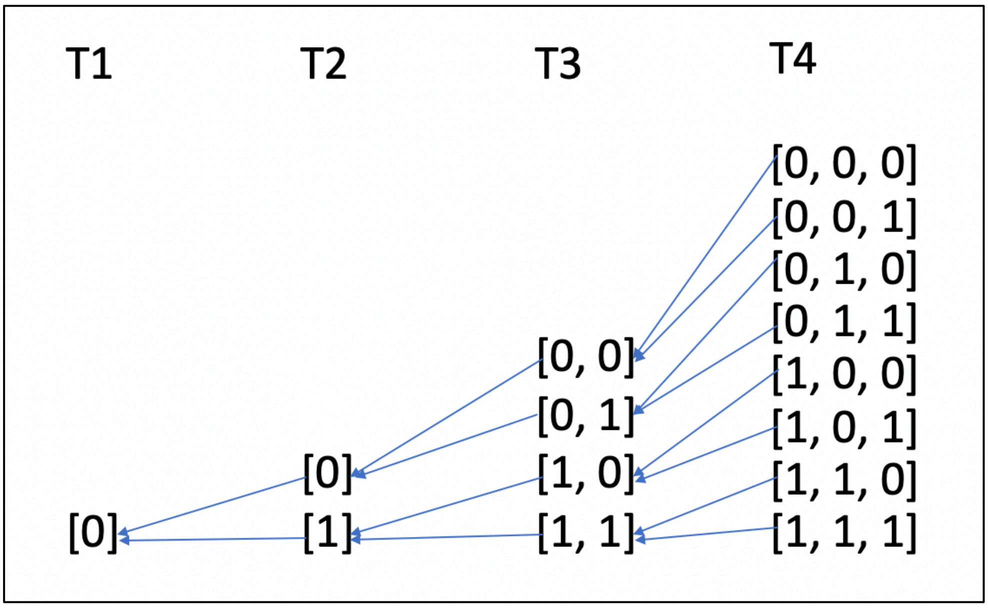
Initial index level, S0, is 4006.18, followed by the barrier level, 3204.944, which is 80% of the initial index level.

Principal amount per security is $1000, which we'll call it face value, pays a coupon of $28.75 per security on the coupon dates.

The pricing date is September 9, 2022. Followed by interim dates, December 22, 2022, March 23, 2023, and June 22, 2023, final valuation date September 21, 2023, and maturity date September 26, 2023. We'll call them T1, T2, T3, and T4 for interim dates and final valuation date respectively. (In the python code, T1, T2, T3, and T4 are the tau values)

Methodology

First, mapping out all the scenarios where from T4 to T3, there are 8 scenarios; from T3 to T2 there are 4 scenarios; from T2 to T1 there are 2 scenarios; and from T1 to T0 there is only one scenario. Second, add an extra dimension to the security value tree when constructing our binomial tree model. We'll call the extra dimension 'k', which represents the number of scenarios. For example, from T4 to T3, k equals to 8, from T3 to T2, k equals to 4, and so on. Then, break down the tree into four parts, from T4 to T3, from T3 to T2, from T2 to T1, and from T1 to T0. At T3, T2, and T1, the scenarios will narrow down by the factor of 2, and then discount along the time steps like normal binomial trees. Finally, the estimated security value will be on the 0^{th} step of the 0^{th} dimension (k = 0). Figure below shows how different combinations of path (scenarios) narrow down on the interim dates.



Algorithms

At T4, we calculate the payoffs for each scenario, k, where if the index level is greater or equal to the barrier level, 0.8*S0, we'll get a payoff equals to principal amount, plus future value of paid coupon on previous coupon dates, plus unpaid contingent coupon, and plus the coupon payment at T4. However, if the index level is below the barrier level, the payoff will be principal amount plus principal amount times the simple return of the index.

$$For\ k\ =\ 0, 1, 2, \dots, 7, and\ j\ =\ T_4$$

$$V_{k,j,i} = \begin{cases} face + FV(paid\ coupon) + unpaid\ coupon + coupon_{T4}\ , if\ S_{T4} \geq 0.8S_0 \\ face + face * \dfrac{(S_{T4} - S_0)}{S_0}, if\ S_{T4} < 0.8 * S_0 \end{cases}$$

$$For\ k\ =\ 0,1,2,\ldots,7, and\ j\ =\ T_4-1, T_4-2,\ldots,T_3+2,T_3+1$$

$$V_{k,j,i}\ =\ e^{-r\Delta t}*(q*V_{k,j+1,i+1}+(1-q)*V_{k,j+1,i})$$

At T3, the number of scenarios will be down to 4, since the possible scenarios at T4 have been realized. Then we check if the spot price of the index is above initial level, in between initial level and barrier level, or below barrier level. If the spot price is above initial level, the security will be called, payoff will be principal amount, plus future value of paid coupon on previous coupon dates, plus unpaid contingent coupon, and plus the coupon payment at T3. If the spot price is in between initial level and barrier level, the payoff will be interpreted from the previous time step of binomial tree model on dimension 2k+1. If the spot price is below barrier level, the payoff will be interpreted from the previous time step of binomial tree model on dimension 2k.

$$For\ k=0,1,2,3, and\ j=T_3$$

$$V_{k,j,i}\begin{cases} face+FV(paid\ coupon)+unpaid\ coupon+coupon_{T3}\ ,if\ S_{T3}\geq S_0 \\ e^{-r\Delta t}*\left(q*V_{2k+1,j+1,i+1}+(1-q)*V_{2k+1,j+1,i}\right), if\ S_0>S_{T3}\geq 0.8S_0 \\ e^{-r\Delta t}*\left(q*V_{2k,j+1,i+1}+(1-q)*V_{2k,j+1,i}\right), if\ S_{T3}<0.8S_0 \end{cases}$$

$$For\ k\ =\ 0,1,2,3, and\ j\ =\ T_3-1,T_3-2,\ldots,T_2+2,T_2+1$$

$$V_{k,j,i}\ =\ e^{-r\Delta t}*(q*V_{k,j+1,i+1}+(1-q)*V_{k,j+1,i})$$

Same things happen to T2, only the number of scenarios will be down to 2 at T2.

$$For\ k=0,1, and\ j=T_2$$

$$V_{k,j,i}\begin{cases} face+FV(paid\ coupon)+unpaid\ coupon+coupon_{T2}\ ,if\ S_{T2}\geq S_0 \\ e^{-r\Delta t}*\left(q*V_{2k+1,j+1,i+1}+(1-q)*V_{2k+1,j+1,i}\right), if\ S_0>S_{T2}\geq 0.8S_0 \\ e^{-r\Delta t}*\left(q*V_{2k,j+1,i+1}+(1-q)*V_{2k,j+1,i}\right), if\ S_{T2}<0.8S_0 \end{cases}$$

$$For\ k\ =\ 0,1, and\ j\ =\ T_2-1,T_2-2,\ldots,T_1+2,T_1+1$$

$$V_{k,j,i}\ =\ e^{-r\Delta t}*(q*V_{k,j+1,i+1}+(1-q)*V_{k,j+1,i})$$

Then, at T1, the number of scenarios will be down to 1 at T1. There will be no paid and unpaid contingent coupon at this period.

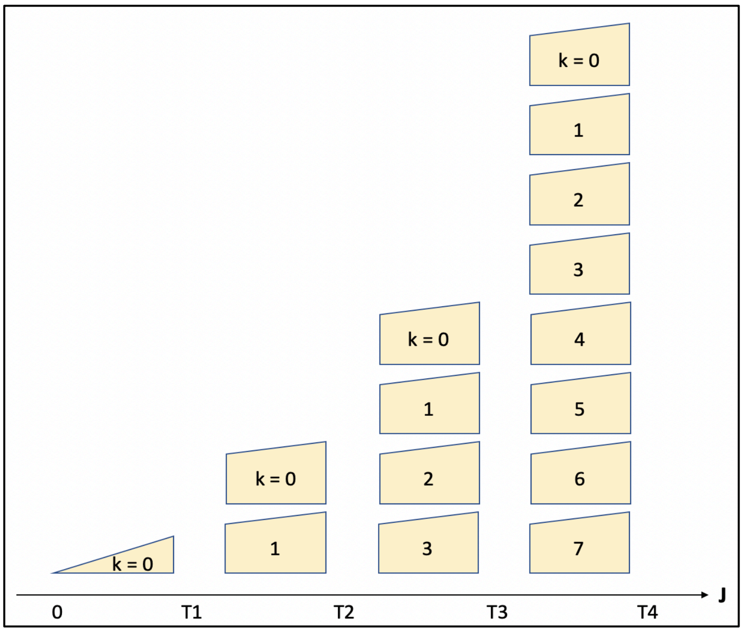
$$For\ k=0, and\ j=T_1$$

$$V_{k,j,i}\begin{cases} face+coupon_{T1}\ ,if\ S_{T1}\geq S_0 \\ e^{-r\Delta t}*\left(q*V_{2k+1,j+1,i+1}+(1-q)*V_{2k+1,j+1,i}\right), if\ S_0>S_{T1}\geq 0.8S_0 \\ e^{-r\Delta t}*\left(q*V_{2k,j+1,i+1}+(1-q)*V_{2k,j+1,i}\right), if\ S_{T1}<0.8S_0 \end{cases}$$

$$For\ k\ =\ 0, and\ j\ =\ T_1 - 1, T_1 - 2, \ldots, 0$$

$$V_{k,j,i}\ =\ e^{-r\Delta t} * (q * V_{k,j+1,i+1} + (1-q) * V_{k,j+1,i})$$

Finally, we will get the estimated value $V_{0,0,0}$ at time step $j = 0$. Graph below shows how methodology is implemented in the algorithms for 3-dimensional binomial tree.



Special Python functions

Cpn_payments(time) is a function that takes the coupon time and calculate the paid and unpaid contingent coupons for that time. To properly calculate the coupons, we first map out the possible path (scenarios) in the form of array. Then, map out the coupons and discount them properly and save them as array, interim_cpn. Finally, the product of the two arrays yields the discounted coupons paid on interim dates for different possible scenarios. To get the unpaid contingent coupons on the valuation dates, we do 1's complement (turn 0 to 1 and 1 to 0) on array 'path', and then create an array of coupons without discounting them. The product of the two arrays yields the unpaid contingent coupons for all possible scenarios.

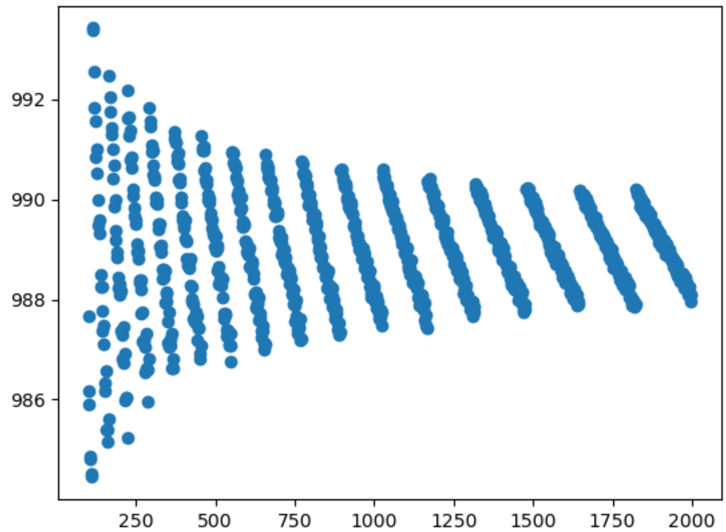
Sensitivity analysis

Running sensitivity analysis on a pool of sigma values (see supplement data) gives the table below, where we can see that the value is closest to the CGMI's estimated value when sigma is equal to 0.23441.

Sigma	Value
0.32036	957.350154
0.30755	959.28134
0.30212	962.447277
0.29587	965.00101
0.21967	994.797044
0.22862	991.397662
0.23319	988.833042
0.23441	988.711803

Non-linearity error

The graph below shows that there might be non-linearity errors in our estimation. We run the simulate security value using 100 to 2000 steps. As a result, we can see that the value of security converges to around $988.90 as time step increases.



Conclusion

The estimated value for this project is $988.71, after sensitivity analysis on the list reasonable volatilities using a time step of 3770. Considering the convergence value, $988.90, as the true value, the implied volatility we get is 0.2335.

Considerations for this project might be insufficient time steps used in the binomial tree, since we know that the error will be much smaller as time step increases. Further, considering the non-linearity analysis, we could have tried much more time steps such as 10,000 steps, to observe a better convergence of the estimated value. However, the computing power was insufficient at this point so we think doing 2,000 steps should be good enough for now.

Supplement data

Volatility source

Risk-free rate

Dividend yields


OVME S&P 500 INDEX Index
Options
S&P 500 INDEX Index
OVME
Related Functions Menu
Message
SPX
3633.02
-111.50
3632.47 /3633.40
At 15:48 d
O 3706.74
H 3706.74
L 3620.73
Prev 3744.52
XD
Asset
Actions
Products
Views
Settings
Option Pricer Equity/IR
New equity and index deal IDs will be 10 characters from 01 Sep. More »
12) Solver (Vol)
13) Load
14) Save
15) Trade
16) Ticket
17) Send
21) Deal 1
22) +
31) Pricing
32) Scenario
33) Matrix
34) Volatility
35) Backtest
Underlying
SPX Index
S&P 500 INDEX
Trade
09/09/2022
16:15
Und. Price
4,006.18
USD
Settle
09/26/2023
Results
Price (Total)
219.68
Currency
USD
Vega
8.65
Time Value
219.68
Price (Share)
219.6765
Delta (%)
51.79
Theta
-0.86
Gearing
18.24
Price (%)
5.4834
Gamma (%)
2.9938
Rho
0.06
Break-Even (%)
5.48
European Vanilla
Leg 1
Style
Vanilla
Exercise
European
Call/Put
Call
Direction
Buy
Strike
4,006.18
Strike
% Money
ATM
Shares
1.00
Expiry
01/05/2023
16:15
Time to Expiry
118
00:00
Model
BS - continuous
Vol
BVOL
Ask
22.302%
Forward
Carry
4,026.79
USD Rate
MMkt
3.310%
Dividend Yield
1.642%
Discounted Div Flow
21.16
Suggested Functions
ECO See upcoming economic releases & events
CDR See calendar trading & settlement dates