Pricing Contingent Income Auto-Callable Securities

Piyush Sen - piyush2 - 676132621

Srikari Rallabandi - srikari3 - 654293920

Aastha Shah - ashah357 - 663579707

May 02, 2025

1 Executive Summary

This report presents the valuation of a 2-year Contingent Income Auto-Callable Note linked to HAL, VLO, and XOM equities. The note pays quarterly coupons if all stocks remain above 50% of their initial prices and auto-calls if all exceed 100% on any observation date. At maturity, losses are realized if any stock breaches its downside barrier.

We implement a Monte Carlo simulation using 1,024,000 paths under a geometric Brownian motion framework with correlated asset dynamics. Inputs include implied volatilities at 100% and 50% moneyness, dividend yields, forward SOFR-based rates, and historical correlations. Valuation is conducted via backward induction across determination dates, capturing path-dependent features such as auto-call, coupon eligibility, and downside protection.

The final estimated note value is $952.91, computed as the average of values under both volatility scenarios. Sensitivity analysis confirms the expected inverse relationship between volatility and note value due to early redemption effects. Convergence analysis supports the robustness and stability of the Monte Carlo approach used.

2 Introduction

2.1 Product Specifications

- **Issuer:** UBS AG London Branch
- **Underlying Equities:**
  - Common stock of Halliburton Company (Bloomberg Ticker: "HAL UN")
  - Common stock of Valero Energy Corporation (Bloomberg Ticker: "VLO UN")
  - Common stock of Exxon Mobil Corporation (Bloomberg Ticker: "XOM UN")
- **Stated Principal Amount:** $1,000.00 per security
- **Issue Price:** $1,000.00 per security
- **Pricing Date:** March 7, 2025
- **Original Issue Date:** March 12, 2025

- **Maturity Date:** March 11, 2027
- **Early Redemption:** The securities will be automatically redeemed on any determination date other than the final determination date if the closing prices of all the underlying equities are equal to or greater than their respective call threshold levels.
- **Early Redemption Amount:** The early redemption amount is equal to the stated principal amount plus any contingent payment otherwise payable with respect to the related determination date.
- **Contingent Payment:**
  - \$25.775 per security (equivalent to 10.31% per annum of the stated principal amount) if the closing prices of all of the underlying equities are equal to or greater than their respective coupon barrier levels on a determination date.
  - No contingent payment if the closing price of any underlying equity is less than its respective coupon barrier level on a determination date.
- **Payment at Maturity:**
  - If the final prices of all of the underlying equities are equal to or greater than their respective downside threshold levels, the payment will be the stated principal amount plus any contingent payment otherwise payable on the maturity date.
  - If the final price of any underlying equity is less than its downside threshold level, a cash payment will be made that is less than the stated principal amount (if any), calculated as:

$$\text{Stated Principal Amount} + (\text{Stated Principal Amount} \times$$

$$\text{Underlying Return of Worst Performing Equity})$$

Asset	Init. Price	Cou. Barr.	Call Thr.	Downside Thr.
HAL	\$25.00	\$12.50 (50%)	\$25.00 (100%)	\$12.50 (50%)
VLO	\$126.85	\$63.43 (50%)	\$126.85 (100%)	\$63.43 (50%)
XOM	\$109.02	\$54.51 (50%)	\$109.02 (100%)	\$54.51 (50%)

Table 1: Initial Prices, Coupon Barrier Level, Call Threshold Level, and Downside Threshold Level for the Underlying Assets

Determination Date	Contingent Payment Date
June 9, 2025	June 12, 2025
September 8, 2025	September 11, 2025
December 8, 2025	December 11, 2025
March 9, 2026	March 12, 2026
June 8, 2026	June 11, 2026
September 8, 2026	September 11, 2026
December 7, 2026	December 10, 2026
March 8, 2027	March 11, 2027 (Maturity)

Table 2: Determination and Coupon Dates

Underlying Return: The underlying return is defined as the percentage change in the price of an underlying equity from its initial price to its final price. It is calculated as:

$$\text{Underlying Return} = \frac{\text{Final Price} - \text{Initial Price}}{\text{Initial Price}} \times 100$$

2.2 Model Brief

The valuation of the auto-callable note is carried out using a Monte Carlo simulation framework that projects stock prices across multiple determination dates and computes payoffs through backward induction. The product features include quarterly contingent coupons and an auto-call feature if all underlyings exceed their initial levels. If the note is not called and any underlying breaches the barrier at maturity, the final payoff reflects the worst-performing stock.

The model is structured in two stages:

2.2.1 Stock Price Simulation

We simulate stock prices under the risk-neutral measure using a geometric Brownian motion process. The process accounts for asset-level dividend yields, volatilities, and correlated shocks derived from historical return data.

The inputs required for this simulation include:

- `S0`: Initial stock prices vector for the three underlyings
- `sigma`: Volatility vector for the underlyings
- `q`: Dividend yield vector
- `r`: Vector of forward risk-free rates between successive determination dates
- `rho`: Historical correlation matrix between the three assets
- `det_dates`: Vector of observation dates expressed in year fractions
- `N`: Total number of Monte Carlo simulation paths

To generate correlated stock paths, we apply Cholesky decomposition to the correlation matrix `rho` to produce a lower triangular matrix `C`, such that:

$$CC^\top = \rho$$

For each time step and asset, we compute the stock price using the discretized GBM formula:

$$S_{i,j,k} = S_{i,j-1,k} \cdot \exp\left[(r_j - q_k - \frac{1}{2}\sigma_k^2)\Delta t_j + \sigma_k \sqrt{\Delta t_j} \cdot Z_{i,j,k}\right]$$

Here:

- $i \in \{1, 2, \ldots, N\}$ indexes the simulation path
- $j \in \{1, 2, \ldots, T\}$ indexes the determination date
- $k \in \{1, 2, \ldots, A\}$ indexes the asset
- $Z_{i,j,k}$ are correlated standard normal shocks generated using $Z = C \cdot \phi$, where $\phi \sim \mathcal{N}(0, 1)$
- Δt_j is the time difference in years between determination dates $j - 1$ and j

The output of the simulation is a 3D array `S` of dimension $(N, \texttt{len(det\_dates)}, 3)$, storing the simulated prices of each underlying at each observation date for all paths.

2.2.2 Valuation Framework

The Monte Carlo valuation proceeds using backward induction:

- On every determination date, continuation value is calculated as the discounted value of the value at the next determination date of the same path
- If all three stocks are above the coupon barrier (50% of initial) at a determination date, the coupon is added to the continuation value
- If all stocks are above the call threshold (100% of initial), the note is auto-called and the investor receives principal + coupon.
- At any point the coupon is paid, it has to be discounted using the forward rate for that time zone and the time lapsed between the determination date and following contingent coupon date.
- While looping on determination dates, discounting is applied using per-period forward rates between two determination dates to estimate the continuation value.
- At maturity, if any stock falls below the barrier, the principal is reduced based on the return of the worst-performing stock.
- If all stock prices are equal to or greater than the respective threshold levels, then the value is the stated principal and contingent payment.

The final value is the discounted value of the average value at time step 0.

This method captures realistic payoff dynamics and path dependencies across multi-asset structures while maintaining model robustness and transparency.

2.3 Data Description

2.3.1 Conventions and Notation

This section outlines the key notations and variables used throughout the simulation and valuation framework.

- `S0`: Vector of initial stock prices for the three underlyings
- `sigma`: Volatility vector for each asset. Volatilities were extracted from Bloomberg for both 100% and 50% moneyness levels and averaged for final use in the model
- `q`: Dividend yield vector for each underlying, used as a continuous yield
- `rho`: Historical correlation matrix across the three assets, computed using Bloomberg's HRA function over the past 6 months
- C is the lower triangular Cholesky factor of the correlation matrix ρ, where $\rho = CC^T$, used to generate correlated random shocks across assets
- `forward_rate`: Forward risk-free rate vector between successive determination and contingent payment dates, derived from SOFR discount factors
- `det_dates`: Vector of quarterly observation dates expressed in year fractions, used as time anchors for simulation
- `cont_dates`: Vector of coupon payment dates, also expressed in year fractions
- `coupon_barrier`: Vector of coupon or downside thresholds, defined as a percentage of the initial stock price
- `call_level`: Vector defining the auto-call threshold for each asset, typically set at 100% of initial value
- `V`: 2D array storing note value for each path and determination date
- `cont_value`: 2D array holding the discounted continuation values for each path and step
- `S`: 3D array storing the simulated stock prices $(N, \texttt{len(det\_dates)}, 3)$
- `discount[i]`: Stores pre-computed discount factors $e^{-r_i \times \Delta t_i}$ for each time interval, where r_i is the forward rate and Δt_i is the time between dates i and $i+1$. These factors are used to calculate continuation values during backward induction.

Discount Factor and Forward rate The discount factor $P(0,T)$ for a future time T is obtained from Bloomberg's SWDF command using SOFR-based rates. Since exact dates were not always available, linear interpolation was used between published discount factors. The global date used for all market data extraction was set to 03/07/2025.

Implied Volatility Note We considered volatilities for both 100% and 50% moneyness levels from Bloomberg's OVME surface, and used the average across the two to capture skew while preserving model tractability. Time to maturity was defined as 2.0 years, aligned with the structure of the note.

Other Parameters All coupon and payoff logic was applied on determination and contingent payment dates. Simulations were carried out for all paths across these dates only.

2.4 Data Collection and Estimation

Besides the information provided in the term sheet, we used Bloomberg to calculate the implied volatility, risk free rate, dividend yield and correlation of the three stocks.

2.4.1 Implied Volatility (σ)

Implied volatilities were extracted from Bloomberg using the OVME command for Halliburton (HAL), Valero (VLO), and Exxon Mobil (XOM). We obtained surface volatilities at both 100% and 50% moneyness levels, corresponding to the auto-call threshold and coupon barrier respectively.

The values used are:

Moneyness	HAL	VLO	XOM
100%	35.354%	32.416%	23.255%
50%	41.170%	38.605%	32.691%

Table 3: Implied Volatility by Stock and Moneyness

Since the call threshold is at 100% and coupon barrier is at 50% moneyness levels, we ran Monte Carlo simulations using volatilities at both these moneyness levels and averaged the prices to arrive at our final value.

The time to maturity was taken to be 2.0 years, in line with the maturity of the note.

2.4.2 Estimation Process

The simulation process focuses exclusively on the quarterly determination dates and contingent payment dates, avoiding daily modeling. The vector `det_dates` contains the observation dates expressed in year fractions, while `cont_dates` represents the respective payment dates.

Time Step Unlike traditional binomial models, the time steps here are not uniform. The time intervals between determination dates vary, and each Δt_j is computed dynamically as:

$$\Delta t_j = t_j - t_{j-1}$$

This ensures accurate discounting and drift computation between irregular time gaps in quarterly periods.

Correlation Matrix To capture realistic cross-asset behavior, we use a historical correlation matrix `rho` computed over the past 6 months using daily log returns of HAL, VLO, and XOM. The correlation values were computed from Bloomberg's HRA function using data from **September 7, 2024 to March 7, 2025**.

Cholesky Decomposition The correlation matrix is transformed via Cholesky decomposition into a lower triangular matrix C such that:

$$\rho = CC^\top$$

This allows us to convert independent standard normal variables into correlated shocks used in the Monte Carlo simulation. Specifically, if $\phi \sim \mathcal{N}(0,1)^3$, then:

$$Z = C \cdot \phi$$

where Z becomes a correlated vector of standard normals used in the GBM diffusion term. The full update to the simulated price uses:

$$S_{i,j,k} = S_{i,j-1,k} \cdot \exp\left[(r_j - q_k - \frac{1}{2}\sigma_k^2)\Delta t_j + \sigma_k \sqrt{\Delta t_j} \cdot Z_{i,j,k}\right]$$

Number of Simulations A total of **1,024,000** paths were used in the Monte Carlo framework. This count was selected to ensure convergence of the valuation and a standard error below \$0.50.

3 Model description

3.1 Monte Carlo specifics

While we initially began by testing the model on 10,000 simulations run 10 times, for the final version we ran 1.024m simulations. The number was chosen based on the achievement of a standard error of less than \$0.50. As discussed in subsequent section, we needed to have at least this many simulations to achieve the desired standard error for both levels of moneyness (100% and 50%).

3.2 Stock Simulation using Monte Carlo

We performed stock price path simulation using the Monte Carlo method based on geometric Brownian motion. This approach allows us to model multiple assets with an underlying correlation structure. The parameters utilized in this construction are defined as:

- S_0: Initial stock price vector for each asset
- σ: Annual volatility vector for each asset
- r: Vector of risk-free forward rates corresponding to each determination date
- q: Dividend yield vector for each asset
- ρ: Correlation matrix between assets
- N: Number of simulation paths
- NA: Number of assets
- det_dates: Vector of determination dates expressed as year fractions

For multiple correlated assets, we employ Cholesky decomposition of the correlation matrix ρ to generate correlated random variables:

$$C = \text{chol}(\rho) \tag{1}$$

Since we are dealing with multiple assets, we need a 3-dimensional array to store the stock prices array with dimensions $[N, \text{det\_dates}, NA]$, representing:

- N is independent price paths

- Determination dates in terms of years
- Asset number - 1/2/3

The simulation proceeds through the following steps:

1. Initialize an array of current prices for each path with the initial stock prices S_0
2. For each path $i \in \{0, 1, \ldots, N-1\}$:
   (a) For each determination date d_idx:
      i. Calculate the time step Δt between current and previous determination date
      ii. Retrieve the appropriate forward rate r_{d_idx} for this period
      iii. Generate standard normal random variables $\phi \sim \mathcal{N}(0, 1)$ for each asset
      iv. Transform into correlated random variables: diffusion_vec $= C \cdot \phi$
      v. For each asset $j \in \{0, 1, \ldots, NA-1\}$:
         A. Generate correlated random shocks: diffusion_vec $= C \cdot \phi$, where $\phi \sim \mathcal{N}(0, 1)$ and C is the Cholesky factor of the correlation matrix
         B. Calculate the drift term: drift_term $= (r_{d\_idx} - q_j - \frac{1}{2}\sigma_j^2) \cdot \Delta t$
         C. Calculate the diffusion term: diffusion_term $= \sigma_j \cdot \sqrt{\Delta t} \cdot \text{diffusion\_vec}_j$
         D. Update the stock price: $S_{i,j} = S_{i,j} \cdot e^{\text{drift\_term}+\text{diffusion\_term}}$
      vi. Store the updated prices in the array $S[i, d, :]$ for all the three assets.

The drift term incorporates the risk-free rate, dividend yield, and volatility adjustment, while the diffusion term captures the random component scaled by volatility and square root of time. This approach is generating a stock price lattice of dimensions (1024000, 8, 3), leading to a total of 24.576m stock prices.

3.3 Valuation

The Monte Carlo valuation approach estimates the autocallable note's value by simulating thousands of possible price paths and applying backward induction. Our approach creates a multidimensional array of paths that represent specific realizations of the underlying price evolution for each path, determination date and asset.

3.3.1 Monte Carlo Implementation Framework

The valuation framework employs a 3D matrix structure with the following essential components:

- S: 3D array of simulated stock prices $[N, \text{len}(\text{det\_dates}), NA]$
- V: 2D array storing note values $[N, \text{len}(\text{det\_dates})]$
- cont_value: 2D array storing continuation values $[N, \text{len}(\text{det\_dates})]$
- N: Number of simulated paths
- det_dates: Vector of determination dates expressed as year fractions
- cont_dates: Vector of contingent payment dates expressed as year fractions
- call_threshold: Vector of respective call threshold values for all the stocks (100% of the initial stock price)

- coupon_barrier/downside_threshold: Vector of respective coupon barrier prices for all the stocks (50% of the initial stock price)

The valuation follows the backward induction principle, but is calculated across paths rather than nodes. The process begins at maturity and works backward through determination dates.

3.3.2 Value at Final Determination Date

Stock condition	**Pricing condition**	**Implementation**
All assets $\geq$ respective coupon barriers	Principal amount plus Contingent Coupon at maturity	Note value is $(prin + coupon) \times e^{-r_T \times \Delta t}$ where Δt is the delay between determination and payment dates and r_T is the forward risk-free rate between the final determination date and maturity
Any asset $<$ its coupon barrier	Principal amount multiplied by the worst-performing asset return	Note value is $prin \times \min(\frac{S_{i,T,0}}{S_{0,0}}, \frac{S_{i,T,1}}{S_{0,1}}, \frac{S_{i,T,2}}{S_{0,2}})$ where i is the path index and T is the final determination date

Table 4: Maturity Payoff Implementation in Monte Carlo

The implementation at maturity checks if all assets are above their respective coupon barriers. For multi-asset products, the worst-performing asset becomes the limiting factor in the downside scenario. The above process will run for all simulation paths from 1 to N.

3.3.3 Backward Induction Through Determination Dates

The backward induction process works from the second-to-last determination date backward to the first, applying the following equation for each path and date:

$$\text{cont\_value}_{n,i} = \text{discount}_{i+1} \times V_{n,i+1} \tag{2}$$

The discount factor (discount_i+1) represents the discounting between consecutive determination dates and is pre-computed based on the forward rate structure. At each determination date, three conditions are evaluated for each path:

Asset condition	Pricing condition	Implementation
All assets $\geq$ respective call thresholds (Auto-call condition)	Cash payment of principal amount plus Contingent Coupon	$V_{n,i} = (prin + coupon) \times e^{-r_i \times \Delta t}$ where Δt is the delay between determination and payment dates and r_i is the forward rate between the determination and coupon payment date
All assets $\geq$ respective coupon barriers but no auto-call	Contingent Coupon payment only	$V_{n,i} = \text{cont\_value}_{n,i} + \text{coupon} \times e^{-r_i \times \Delta t}$
Any asset $<$ its coupon barrier	No payment, note continues	$V_{n,i} = \text{cont\_value}_{n,i}$

Table 5: Determination Date Conditions in Monte Carlo

The auto-call feature introduces path-discontinuities in the valuation process. When all assets exceed their call thresholds, the path effectively terminates with a known payoff, thus reducing potential returns. Intuitively, we expect that if the implied volatility we assume is higher, asset prices may more paths having higher than call threshold prices and thus leading to a lower price of the note due to exercise of the auto call feature. We will investigate the results from different implied volatilities (arising from different moneyness levels).

3.3.4 Final Valuation and Error Estimation

The final note value is the average of path values at time zero, discounted from the first determination date:

$$\text{Note Value} = \frac{1}{N} \sum_{n=1}^{N} V_{n,0} \times \text{discount}[0] \tag{3}$$

A key advantage of Monte Carlo over binomial approaches is the ability to estimate the standard error of the valuation. This standard error calculation involves the following steps. We compute the standard deviation of these path values, $\sigma_{\text{path values}}$.

Then, we calculate the standard error as the standard deviation divided by the square root of the number of paths:

$$\text{Standard Error} = \frac{\sigma_{\text{path values}}}{\sqrt{N}} \tag{4}$$

This standard error decreases at a rate proportional to $1/\sqrt{N}$, indicating that quadrupling the number of simulation paths halves the standard error. Results from the convergence of increasing simulations by a factor of 4 is shown in subsequent sections (should lead to reducing standard error by 2).

3.4 Execution of the Stock Price Path and Valuation

The implementation of our Monte Carlo valuation framework includes the following components:

1. **Market Parameters:** Stock prices, volatilities, dividend yields, risk-free rates, correlation matrices, and Cholesky decomposition are implemented as shown in the prior sections.

2. **Simulation Calibration:** The optimal number of simulations was determined by analyzing the convergence of standard error. We established a threshold of \$0.50 for the standard error and conducted convergence testing at both 100% and 50% moneyness levels. Our analysis indicated that $N = 1,024,000$ simulations would be required to achieve a standard error below \$0.50 for both moneyness levels.

3. **Stock Price Generation:** The `MC_StockPrices` function generates correlated stock price paths for multiple assets using geometric Brownian motion with the specified parameters. This returns a 3D stock array with the dimensions being the path, determination date and the asset.

4. **Valuation Function:** The `value` function implements backward induction through the determination dates, evaluating early redemption conditions and calculating continuation values for each path. This function first determines the value of the note on each path on maturity based on the stock prices of the three assets. Thereafter it iterates through each of the remaining 7 determination dates and checks for early redemption and coupon barrier levels. The final value is essentially the mean of the value at the first determination date from all the paths and multiplied by the discount factor between T=0 and the first determination date.

5. **Simulation Execution:** Each simulation of 1024000 paths was run 10 times to arrive at the final value for the value by taking a mean of those 10 values. This process was repeated for both the moneyness levels and the final value was the mean of those two values.

6. **Results:** For a 100% moneyness, we obtained a value of \$980.98, while for 50% moneyness we obtained a value of \$924.83. The final value was **\$952.91**.

7. **Error Analysis:** The convergence of standard error was plotted against the number of simulations to verify the $O(1/\sqrt{N})$ theoretical convergence rate, confirming the statistical validity of our approach.

4 Discussion

4.1 Results Analysis

Based on our model, we arrive at the following results.

1. At a moneyness of 100% and N = **1,024,000** simulations, we obtained a value of \$980.98. Below is the distribution of the note prices at 100%.

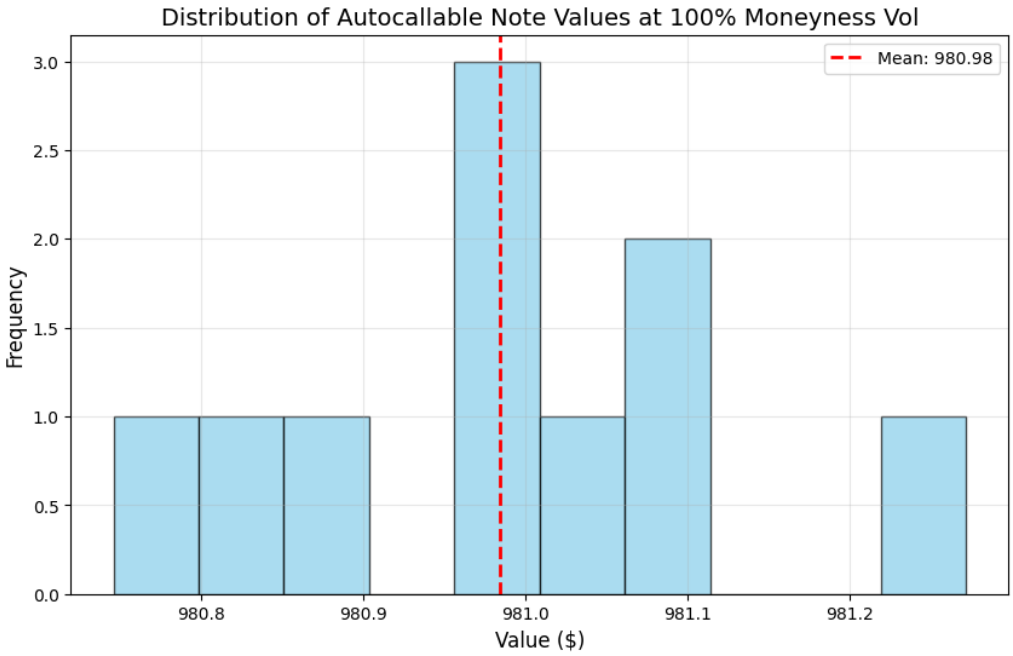

Figure 1: Price distribution at 100% moneyness

2. At 50% moneyness levels, we obtained a value of $924.83. Below is the distribution of the prices.

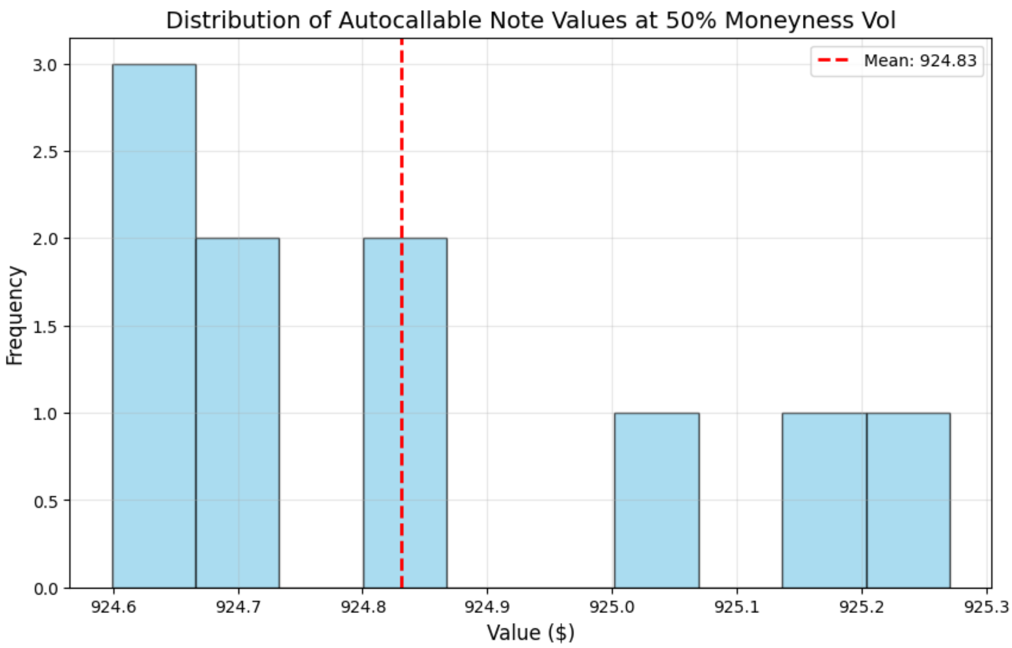

Figure 2: Price distribution at 50% moneyness

The final note price is estimated as the average of these two as **$952.91**.

4.2 Sensitivity Analysis

Below we present the sensitivity analysis of the note price with respect to two key parameters: volatility and simulations.

4.2.1 Volatility Sensitivity

As we have shown above, we have shown the value at 100% and 50% moneyness levels of implied volatility for the different stocks.

Company Name	Implied Volatility (100%)	Implied Volatility (50%)
Halliburton	35.354	41.17
Valero Energy	32.416	38.605
Exxon Mobil	23.255	32.691
Final Note Price	**$980.98**	**$924.83**

Table 6: Implied Volatilities and Note Prices

As we note, all three stocks have higher implied volatility at 50% moneyness levels. Lower implied volatility leads to lower probability of stock prices hitting the call threshold levels thus resulting in lower chances of getting auto called. This means more paths on any given determination date are falling either in the coupon payment stage or pure continuation stage, which by virtue of their nature will contribute to a higher value. A higher volatility, on the other hand, leads to more paths hitting the call threshold levels and thus are getting auto called, thereby capping the value. Hence, higher volatility leads to a lower note value.

4.2.2 Convergence

Table 7: Convergence Analysis for 50% Moneyness

Simulations	Price ($)	Std Error ($)	Time (s)
1,000	938.44	7.9862	0.17
2,000	920.69	5.9203	0.40
4,000	927.50	4.1083	0.70
8,000	928.98	2.9174	1.70
16,000	925.57	2.0656	3.79
32,000	925.64	1.4647	6.23
64,000	926.68	1.0324	13.30
128,000	925.39	0.7333	26.75
256,000	924.67	0.5192	54.97
512,000	925.52	0.3663	96.83
1,024,000	925.03	0.2593	194.74

Based on the convergence analysis data for 50% and 100% moneyness levels, we identify the following key inferences:

Table 8: Convergence Analysis for 100% Moneyness

Simulations	Price ($)	Std Error ($)	Time (s)
1,000	975.01	6.9583	0.23
2,000	986.21	4.6723	0.33
4,000	981.71	3.4118	0.99
8,000	980.50	2.4109	1.63
16,000	980.46	1.6993	3.09
32,000	981.68	1.1898	7.12
64,000	981.73	0.8419	14.00
128,000	980.79	0.5999	26.40
256,000	981.37	0.4223	53.13
512,000	981.15	0.2990	97.06
1,024,000	981.34	0.2112	194.01

- **Convergence Pattern:** Both moneyness levels demonstrate consistent convergence as simulation size increases, with the 50% moneyness price settling around $925 and the 100% moneyness price around $981. This indicates the Monte Carlo approach is stable and reliable for pricing these complex structured products.
- **Standard Error Reduction:** The standard error decreases approximately with the square root of the simulation size ($1/\sqrt{N}$), confirming theoretical expectations. At 1,024,000 simulations, both moneyness levels achieve a standard error below $0.30, providing high confidence in the price estimates.
- **Computational Efficiency:** The time required scales linearly with simulation size, approximately doubling with each doubling of simulations. This linear scaling suggests the computational approach is efficiently implemented without unexpected overhead.

These results validate our Monte Carlo implementation and provide confidence in the accuracy of our pricing model for autocallable structured products under different moneyness scenarios.

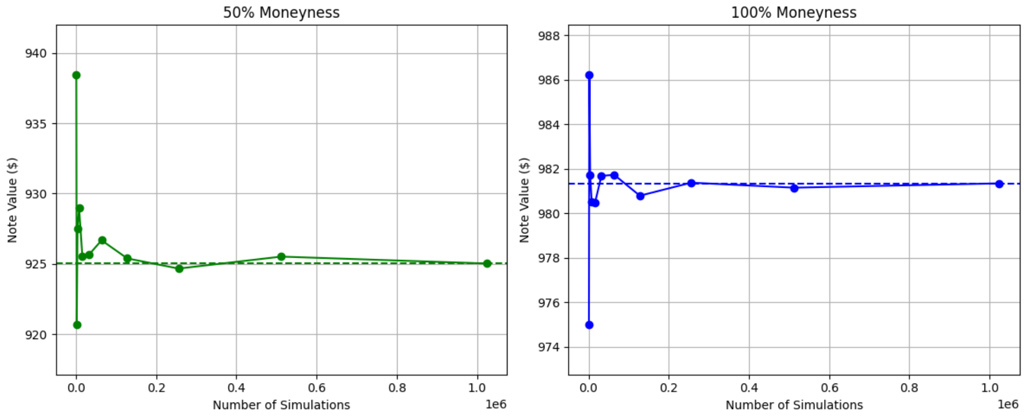

Figure 3: Convergence of prices by varying simulations

5 Appendix

Date	Market Rate	Shift (bp)	Shifted Rate	Zero Rate	Discount
03/18/2025	4.33145	+0.00000	4.33145	4.38976	0.999158
03/25/2025	4.33100	+0.00000	4.33100	4.38746	0.998319
04/01/2025	4.33100	+0.00000	4.33100	4.38562	0.997480
04/11/2025	4.33200	+0.00000	4.33200	4.38399	0.996284
05/12/2025	4.33270	+0.00000	4.33270	4.37657	0.992593
06/11/2025	4.31425	+0.00000	4.31425	4.35023	0.989095
07/11/2025	4.27890	+0.00000	4.27890	4.30718	0.985707
08/11/2025	4.24125	+0.00000	4.24125	4.26186	0.982294
09/11/2025	4.20710	+0.00000	4.20710	4.22032	0.978950
12/11/2025	4.09680	+0.00000	4.09680	4.09003	0.969655
03/11/2026	4.00975	+0.00000	4.00975	3.98498	0.960934
09/11/2026	3.86150	+0.00000	3.86150	3.85038	0.943731
03/11/2027	3.80272	+0.00000	3.80272	3.77912	0.927203
03/13/2028	3.74978	+0.00000	3.74978	3.72659	0.893951
03/12/2029	3.74023	+0.00000	3.74023	3.71796	0.861636
03/11/2030	3.74409	+0.00000	3.74409	3.72288	0.830070
03/11/2031	3.75680	+0.00000	3.75680	3.73704	0.799056
03/11/2032	3.77055	+0.00000	3.77055	3.75227	0.768846
03/11/2033	3.78418	+0.00000	3.78418	3.76748	0.739630
03/13/2034	3.79697	+0.00000	3.79697	3.78180	0.711218
03/12/2035	3.81294	+0.00000	3.81294	3.80007	0.683643
03/11/2037	3.84751	+0.00000	3.84751	3.84035	0.630553
03/12/2040	3.88730	+0.00000	3.88730	3.88765	0.557842
03/13/2045	3.89496	+0.00000	3.89496	3.89247	0.458754
03/11/2050	3.84005	+0.00000	3.84005	3.80627	0.385894
03/11/2055	3.76157	+0.00000	3.76157	3.67981	0.331327
03/11/2065	3.58544	+0.00000	3.58544	3.38662	0.257799
03/11/2075	3.40466	+0.00000	3.40466	3.06995	0.215244

References

1. Bloomberg documentation: https://www.bloomberg.com/professional/support/documentation/

2. Term Sheet: https://www.sec.gov/Archives/edgar/data/1114446/000183988225015118/ubs_-424b2-07602.htm