**Submitted by: Sharang Mantri** **Submitted To: Prof Martin Widdicks**

## Executive Summary

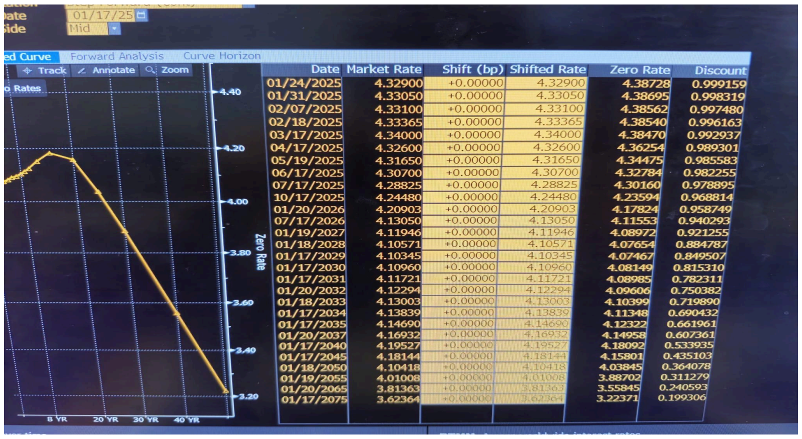
This report presents the valuation of the Dual-Directional Buffered PLUS linked to the S&P 500 Index using static replication. Using Put options and zero coupon bond to value our product via a modified version of the Black Scholes formula we estimated the value of the product to be $962.35 using data of implied volatility, dividend yield, and risk-free rate collected via Bloomberg terminal.
We also conducted a sensitivity analysis using implied volatility for selling a put option and buying a put option being considered different and calculated the value of the product to be $964.90.
As part of the sensitivity analysis, we varied value of all implied volatilities from 1.25 times to 0.75 times of initial value and found product pricing varying from $976 to $1008
The same analysis varying dividend from 1.25 times to 0.75 times of initial value gave a range of product valuation from $958 to $965.

## Data Collection and Sources

The valuation is based on the following data sources:

1) Discount rates using Bloomberg command SWDF and then USD OIS to calculate OIS for the dates 1/24/2025, 1/31/2025, 1/19/2027,1/18/2028 and then used the weightage average of the discount rates between the available future discount rate dates to calculate r(T0, T2): Pricing range of the product, r(T1, T3): Cash flow discounting of the product.

| | Date | Discount (dsi) |
|---|---|---|
| d1 | 1/24/2025 | 0.999159 |
| d2 | 1/31/2025 | 0.998319 |
| d3 | 1/19/2027 | 0.921255 |
| d4 | 1/18/2028 | 0.884787 |



| Date | Market Rate | Shift (bp) | Shifted Rate | Zero Rate | Discount |
|---|---|---|---|---|---|
| 01/24/2025 | 4.32900 | +0.00000 | 4.32900 | 4.38728 | 0.999159 |
| 01/31/2025 | 4.33050 | +0.00000 | 4.33050 | 4.38695 | 0.998319 |
| 02/07/2025 | 4.33100 | +0.00000 | 4.33100 | 4.38562 | 0.997480 |
| 02/18/2025 | 4.33365 | +0.00000 | 4.33365 | 4.38540 | 0.996163 |
| 03/17/2025 | 4.34000 | +0.00000 | 4.34000 | 4.38470 | 0.992937 |
| 04/17/2025 | 4.32600 | +0.00000 | 4.32600 | 4.36254 | 0.989301 |
| 05/19/2025 | 4.31650 | +0.00000 | 4.31650 | 4.34475 | 0.985583 |
| 06/17/2025 | 4.30700 | +0.00000 | 4.30700 | 4.32784 | 0.982255 |
| 07/17/2025 | 4.28825 | +0.00000 | 4.28825 | 4.30160 | 0.978895 |
| 10/17/2025 | 4.24480 | +0.00000 | 4.24480 | 4.23594 | 0.968814 |
| 01/20/2026 | 4.20903 | +0.00000 | 4.20903 | 4.17824 | 0.958749 |
| 07/17/2026 | 4.13050 | +0.00000 | 4.13050 | 4.11553 | 0.940293 |
| 01/19/2027 | 4.11946 | +0.00000 | 4.11946 | 4.08972 | 0.921255 |
| 01/18/2028 | 4.10571 | +0.00000 | 4.10571 | 4.07654 | 0.884787 |
| 01/17/2029 | 4.10345 | +0.00000 | 4.10345 | 4.07467 | 0.849507 |
| 01/17/2030 | 4.10960 | +0.00000 | 4.10960 | 4.08149 | 0.815310 |
| 01/17/2031 | 4.11721 | +0.00000 | 4.11721 | 4.08985 | 0.782311 |
| 01/20/2032 | 4.12294 | +0.00000 | 4.12294 | 4.09606 | 0.750382 |
| 01/18/2033 | 4.13003 | +0.00000 | 4.13003 | 4.10399 | 0.719890 |
| 01/17/2034 | 4.13839 | +0.00000 | 4.13839 | 4.11348 | 0.690432 |
| 01/17/2035 | 4.14690 | +0.00000 | 4.14690 | 4.12322 | 0.661961 |
| 01/20/2037 | 4.16932 | +0.00000 | 4.16932 | 4.14958 | 0.607361 |
| 01/17/2040 | 4.19527 | +0.00000 | 4.19527 | 4.18092 | 0.533935 |
| 01/17/2045 | 4.18144 | +0.00000 | 4.18144 | 4.15801 | 0.435103 |
| 01/18/2050 | 4.10418 | +0.00000 | 4.10418 | 4.03845 | 0.364078 |
| 01/19/2055 | 4.01008 | +0.00000 | 4.01008 | 3.88702 | 0.311279 |
| 01/20/2065 | 3.81363 | +0.00000 | 3.81363 | 3.55845 | 0.240593 |
| 01/17/2075 | 3.62364 | +0.00000 | 3.62364 | 3.22371 | 0.199306 |

Calculations for the r(T0, T2): T2-d3 = 10 and d4-T2 = 354
Weighted average discount rate T2 : ds3*(354/364) + ds4*(10/364) = 0.92053132 = dt2
T2 - T0 = 742 days

**r (T0, T2) = - (365/(T2-T0)) * ln(dt2) = r1 = 4.08812315%**

For r(T1, T3), since it is a forward rate, we calculated r(T0, T1) and r(T0, T3) initially and then used those rates to calculate r(T1,T3)

Calculations for the r(T0, T3): T3-d3 = 15 and d4-T3 = 349
Weighted average discount rate T2 : ds3*(349/364) + ds4*(15/364) = 0.91975219 = dt3
T3 - T0 = 747 days

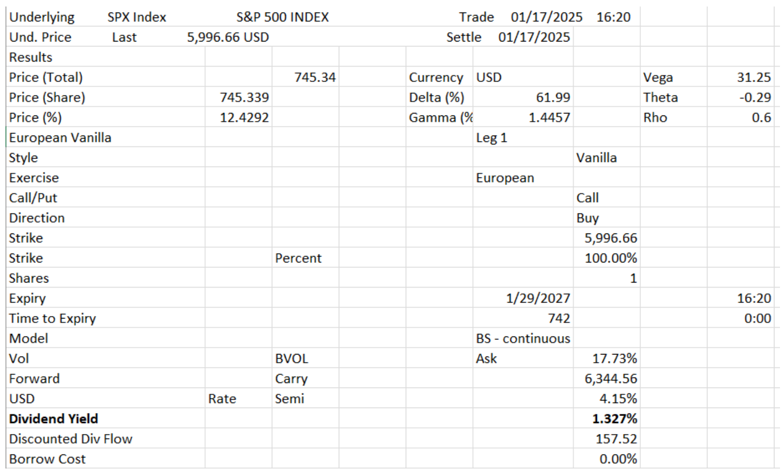
r (T0, T3) = - (365/(T3-T0)) * ln(dt3) = 4.0873646%

Since there was no discount rate available between T0 and T1, we assumed r(T0, d1) = r(T0, T1) due to proximity between T1 and d1
T1 - T0 = 6
d1 - T0 = 7

r(T0, T1) = r(T0, d1) = - (365/(d1-T0)) * ln(ds1) = 4.3870593%
then, T3 - T1 = 741

**r (T1, T3) = [ r(T0, T1) * (T1 -T0) + r(T0, T3) * (T3 - T0) ] / (T3-T1) = 4.15598339% = r2**

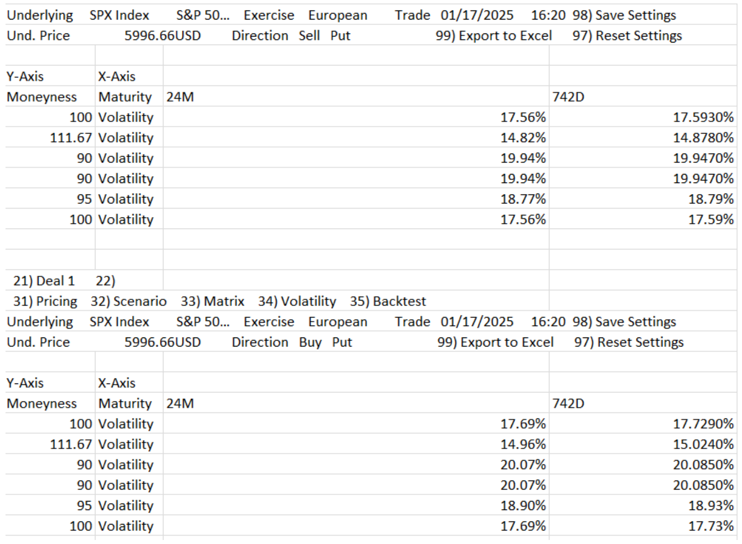
2) For dividend yield between T0 ( Pricing Date) and T2 (Expiry Date), we used Bloomberg command S&P 500 INDEX Index OPDF and collected dividen yield = 1.327%



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Underlying | SPX Index | S&P 500 INDEX | | Trade | 01/17/2025 | 16:20 | | |
| Und. Price | Last | 5,996.66 USD | | Settle | 01/17/2025 | | | |
| Results | | | | | | | | |
| Price (Total) | | 745.34 | | Currency | USD | | Vega | 31.25 |
| Price (Share) | 745.339 | | | Delta (%) | 61.99 | | Theta | -0.29 |
| Price (%) | 12.4292 | | | Gamma (% | 1.4457 | | Rho | 0.6 |
| European Vanilla | | | | | Leg 1 | | | |
| Style | | | | | | Vanilla | | |
| Exercise | | | | | European | | | |
| Call/Put | | | | | | Call | | |
| Direction | | | | | | Buy | | |
| Strike | | | | | | 5,996.66 | | |
| Strike | | Percent | | | | 100.00% | | |
| Shares | | | | | | 1 | | |
| Expiry | | | | | 1/29/2027 | | | 16:20 |
| Time to Expiry | | | | | 742 | | | 0:00 |
| Model | | | | | BS - continuous | | | |
| Vol | | BVOL | | | Ask | 17.73% | | |
| Forward | | Carry | | | | 6,344.56 | | |
| USD | Rate | Semi | | | | 4.15% | | |
| **Dividend Yield** | | | | | | **1.327%** | | |
| Discounted Div Flow | | | | | | 157.52 | | |
| Borrow Cost | | | | | | 0.00% | | |

3) For Implied Volatilities (IVs), we used Moneyness, and maturity matrix based on underlying S&P500 European style. We used Bloomberg command S&P 500 INDEX Index OVM and extracted the IVs with moneyness S0, 1.11667 S0, 0.9 S0 for time period of 742 days (T2 - T0) starting from T0 for buying and selling the put options on the given moneyness and expiring on T2.

| | Index | IVs Put Buy | IVs Put Sell |
|---|---|---|---|
| S0 | 5996.66 | 17.7290% | 17.5930% |
| 0.9S0 | 5396.99 | 20.0850% | 19.9470% |
| 1.11667S0 | 6696.290322 | 15.0240% | 14.8780% |



| | | | |
|---|---|---|---|
| Underlying | SPX Index S&P 50... Exercise European | Trade 01/17/2025 16:20 | 98) Save Settings |
| Und. Price | 5996.66USD Direction Sell Put | 99) Export to Excel | 97) Reset Settings |
| | | | |
| Y-Axis | X-Axis | | |
| Moneyness | Maturity | 24M | 742D |
| 100 | Volatility | 17.56% | 17.5930% |
| 111.67 | Volatility | 14.82% | 14.8780% |
| 90 | Volatility | 19.94% | 19.9470% |
| 90 | Volatility | 19.94% | 19.9470% |
| 95 | Volatility | 18.77% | 18.79% |
| 100 | Volatility | 17.56% | 17.59% |
| | | | |
| | | | |
| 21) Deal 1 | 22) | | |
| 31) Pricing | 32) Scenario 33) Matrix 34) Volatility 35) Backtest | | |
| Underlying | SPX Index S&P 50... Exercise European | Trade 01/17/2025 16:20 | 98) Save Settings |
| Und. Price | 5996.66USD Direction Buy Put | 99) Export to Excel | 97) Reset Settings |
| | | | |
| Y-Axis | X-Axis | | |
| Moneyness | Maturity | 24M | 742D |
| 100 | Volatility | 17.69% | 17.7290% |
| 111.67 | Volatility | 14.96% | 15.0240% |
| 90 | Volatility | 20.07% | 20.0850% |
| 90 | Volatility | 20.07% | 20.0850% |
| 95 | Volatility | 18.90% | 18.93% |
| 100 | Volatility | 17.69% | 17.73% |

# Static replication of the Product

For static replication, we will use all IVs Put Buy and incorporate IVs Put Sell while discussing the sensitivity of the model.

We will start by mentioning some annotations and data:

Leverage factor (LF): 150%

The quantity needed for replication of a certain part of the portfolio will be equal to the slope of that part + slope of the earlier option to balance

Sign of slopes to be determined via the position in option i.e. Long or Short position

For the replication, we will move from Right to Left and use put and zero coupon bond to replicate the product

| | |
|---|---|
| T2 - T0 (in years) = 2.032876712 | Pricing Period |
| T3 - T1 (in years) = 2.030136986 | Discounting Period |
| r1 = 4.08812315% | Pricing Period Risk-Free Rate |
| r2 = 4.15598339% | Discounting Period Risk-Free Rate |

sigma = Implied volatility for the given K
d = Dividend Yield = 1.327%
S0 = 5996.66
0.9S0 = 5396.99
1.11667S0 = 6696.290322
Agent Commission: $25

All initial pricing will be done from T0 with maturity at T2

Replicating Strategy:

1) Invest in a zero coupon bond with FV = $1175 on T0 to get a payoff of $1175 on maturity which is equal to the maximum payoff of the product as we move from Right to Left

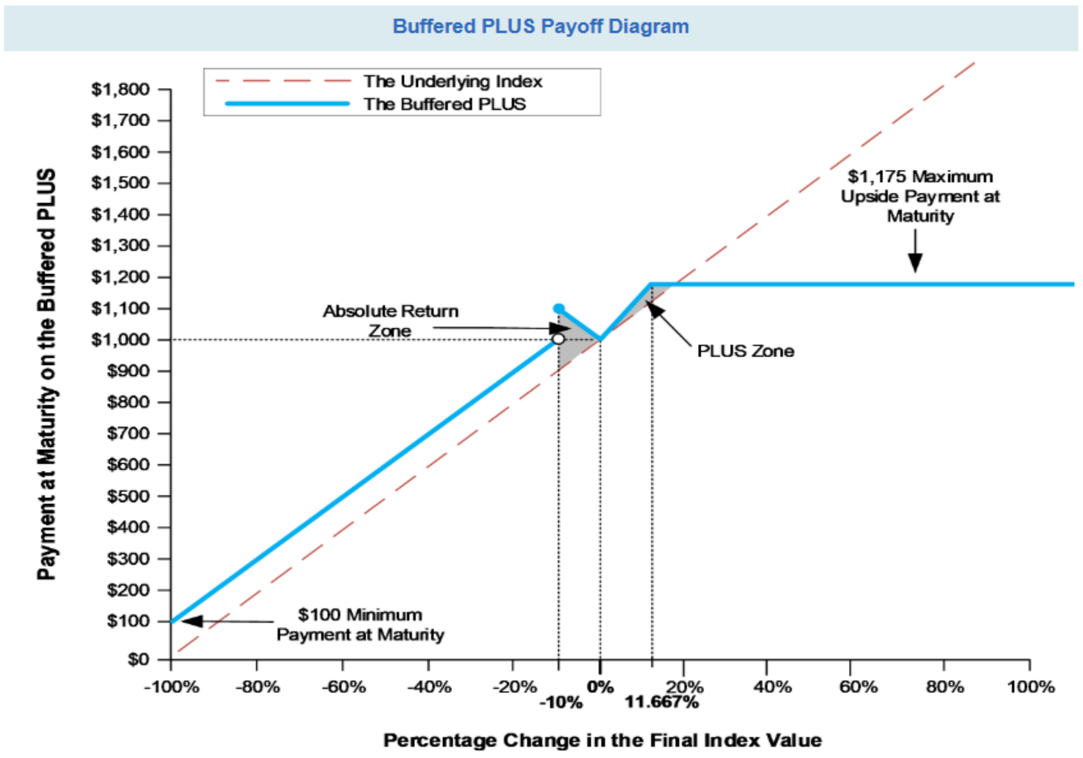
2) Short a Put with K = 1.11667 * S0 with slope LF * 1000/S0 = 0.250139244, to create a negative payoff on if the value of underlying goes below K = 1.11667 S0

3) Long a Put with K = S0 with slope 1000/S0 + LF * 1000/S0 ( To balance previous Short put) i.e. 1000/S0 * (1+LF) = 0.41689874, to create a positive payoff if the value of underlying goes below K = S0.

4) Here in graph, we can clearly see a sudden downside movement at 0.9S0 which requires the use of Digital Option
Short a Digital Put with K=0.9S0 with Payoff = $100 (Drop in the payoff graph at 0.9S0),

to create a positive negative if the value of underlying goes below K = 0.9 S0.

5) Short a Put with K = 0.9 * S0 with slope = 2* 1000/S0 = 0.333518992 to balance long put at K = S0 and create a net short position, to create a negative payoff if the value of underlying goes below K = 0.9 S0.



1) To calculate the value of the PV of bond we used the formula = $FV * e^{-r2*(T3-T1)} = \$1079.93$

For the valuation of Put options, We will use Black Scholes Formula with the following annotation

d1 = $(ln\,(S0 / K) + (r1 - d + sigma^2)/2) * (T2 - T0)) / (sigma * \sqrt{(T2 - T0)})$

d2 = $d1 - (sigma * \sqrt{(T2 - T0)})$

Value of Put = $e^{-r2*(T3-T1)} * (K * N(-d2) - S0 * e^{(r1-d)*(T2-T0)} * N(-d1))$

Value of Digital Option with payoff A = $e^{-r2*(T3-T1)} * A * N(-d2)$

Here, we used the modified version of Black Sholes because we are pricing the  product between different time period while receiving and giving cash back between different time period, The valuation of the expected payoff will be between T0 and T2 while it will be discounted between T3 and T1 with risk free rates associated between the given periods

2) Value of Put at K = 1.11667 S0 = $689.72
   Slope of option (LF * 1000/S0 ) * Value of Option = **$172.53**

3) Value of Put at K = S0 = $425.14
   Slope of option [(1+LF) * 1000/S0 ] * Value of Option = **$177.24**

4) Value of Digital Put at K = 0.9 S0, A=100, S0 = **$30.97**

5) Value of Put at K = 0.9 S0 = $273.83
   Slope of option (2 * 1000/S0 ) * Value of Option = **$91.33**

**Value of the Product = $ ( 1079.93 - 172.53  + 177.24 - 30.97 - 91.33) =  $962.35**

Cost including Agent commission = $962.35 + $25 = **$987.35**

## Sensitivity Analysis

We have made various assumptions while performing the valuation of our product which can lead to certain errors and cause sensitivity in the pricing of the product:

1) We assumed that r(T0,T1) = r(T0, d1) because of no risk free data available between T0 and T1, though this assumption is fairly cautious as risk free rate do not affect the value of option by large amount when compared to other variables and since d1 - t1 = 1, we are pretty certain that this assumption will not cause much noise and thus sensitivity in the value of product.

2) While calculating the value of product, we used all IVs of Buy put with European style option and got valuation of product which is equal to $962.35, if we use IV Buy put only for when we are going long on Put and IV Sell Put when we are going Short on Put we received a valuation of product = $964.90

   Creating a delta of $2.55, which when dealing in large quantities of the product can produce a significant loss.

3) For the valuation, we have also assumed that the dividend yield for our underlying will be 1.327% for the period from the data provided by Bloomberg, while the given yield is

based on historical trends and is true more often than not, but under the certain condition there is a possibility that in real-time dividend yield can be larger or smaller for the same because systematic market risk which can affect the real time dividend and thus can lead to overpaying for the product.

In our sensitivity Analysis, we found if the dividend yield becomes 1.25 times of initial yield value of the product decreases to $958.87 while a yield going to 0.75 times of initial takes the value of the product to $965.78

All these different values can lead to mispricing and overpaying for the product and lead to loss for the investor

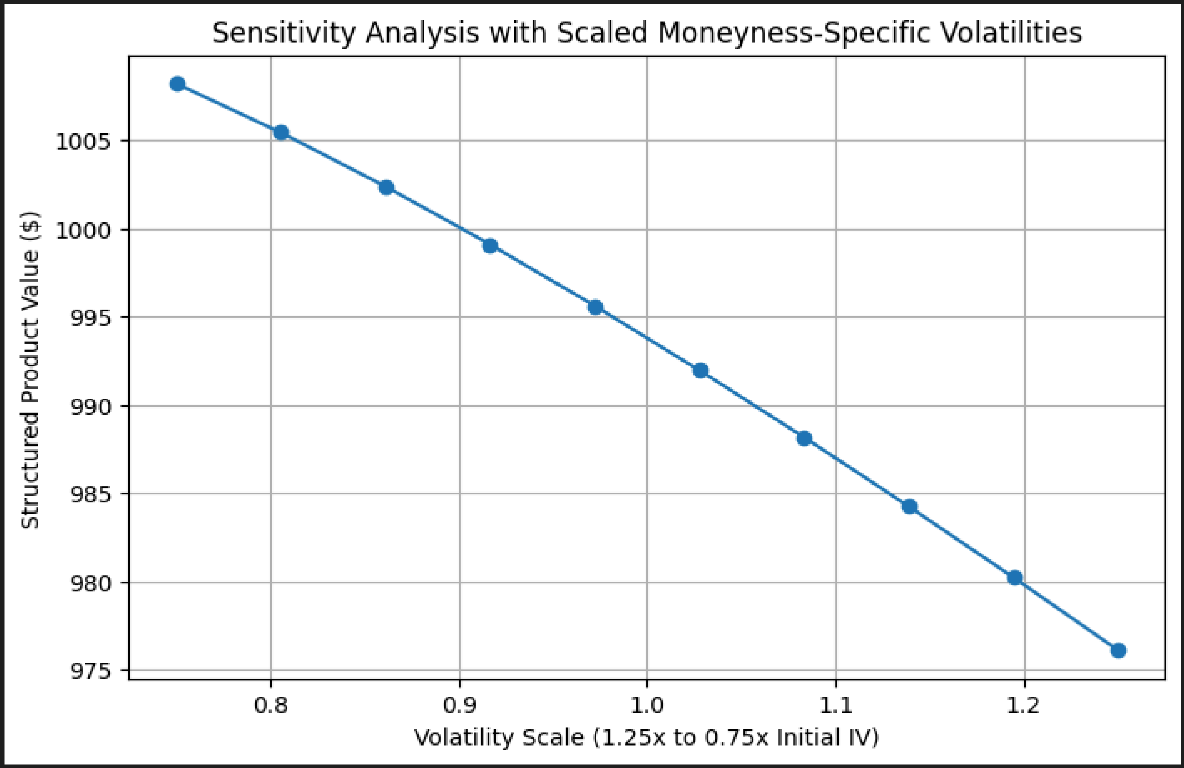
## Appendix:

- Uploaded MS-Excel has built-in VBA of all the valuations as well as data

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.stats import norm

# Parameters from Static Replication
r_pricing = 0.0408812315  # Pricing risk-free rate (T0 to T2)
r_discount = 0.0415598339  # Discounting risk-free rate (T1 to T3)
T_pricing = 2.032876712  # Pricing period (T0 to T2 in years)
T_discount = 2.030136986  # Discounting period (T1 to T3 in years)
S0 = 5996.66  # Initial S&P 500 index level
K_cap = 1.1167 * S0  # Cap strike
K_buffer = 0.9 * S0  # Buffer strike
q = 0.01327  # Dividend yield (1.327%)
LF = 1.5  # Leverage Factor

# Initial Implied Volatilities from Bloomberg for each moneyness
sigma_atm_init = 0.17729  # ATM (S0)
sigma_cap_init = 0.15024  # Cap (1.1167 S0)
sigma_buffer_init = 0.20085  # Buffer (0.9 S0)

# Black-Scholes formula for put option with different rates
def black_scholes_put(S, K, T_price, T_discount, r_price, r_discount, q,
sigma):
    d1 = (np.log(S / K) + (r_price - q + 0.5 * sigma**2) * T_price) /
(sigma * np.sqrt(T_price))
    d2 = d1 - sigma * np.sqrt(T_price)
    return K * np.exp(-r_discount * T_discount) * norm.cdf(-d2) - S *
np.exp(-q * T_price) * norm.cdf(-d1)

# Structured product valuation with scaled moneyness-specific volatilities
def compute_valuation(sigma_atm, sigma_cap, sigma_buffer):
    bond_value = 1175 * np.exp(-r_discount * T_discount)

    put_cap = black_scholes_put(S0, K_cap, T_pricing, T_discount,
r_pricing, r_discount, q, sigma_cap)
    put_atm = black_scholes_put(S0, S0, T_pricing, T_discount, r_pricing,
r_discount, q, sigma_atm)
    digital_put = black_scholes_put(S0, K_buffer, T_pricing, T_discount,
r_pricing, r_discount, q, sigma_buffer)

    total_value = bond_value - (LF * 1000 / S0) * put_cap \
                  + ((1 + LF) * 1000 / S0) * put_atm \
                  - 2 * (1000 / S0) * digital_put

    return total_value

# Sensitivity Analysis with Scaled Volatility Ranges
scale_range = np.linspace(1.25, 0.75, 10)
valuation_results = []
for scale in scale_range:
    sigma_atm = sigma_atm_init * scale
    sigma_cap = sigma_cap_init * scale
    sigma_buffer = sigma_buffer_init * scale

    value = compute_valuation(sigma_atm, sigma_cap, sigma_buffer)
    valuation_results.append(value)

# Display Results

sensitivity_df = pd.DataFrame({
    'Volatility Scale': scale_range,
    'Structured Product Value ($)': valuation_results
})
```

```
print(sensitivity_df)
# Plot Results
plt.figure(figsize=(8,5))
plt.plot(scale_range, valuation_results, marker='o')
plt.xlabel('Volatility Scale (1.25x to 0.75x Initial IV)')
plt.ylabel('Structured Product Value ($)')
plt.title('Sensitivity Analysis with Scaled Moneyness-Specific
Volatilities')
plt.grid(True)
plt.show()
```

```
   Volatility Scale  Structured Product Value ($)
0          1.250000                    976.104825
1          1.194444                    980.219802
2          1.138889                    984.242149
3          1.083333                    988.157659
4          1.027778                    991.949501
5          0.972222                    995.597603
6          0.916667                    999.077869
7          0.861111                   1002.361175
8          0.805556                   1005.412087
9          0.750000                   1008.187249
```


Sensitivity Analysis with Scaled Moneyness-Specific Volatilities
Structured Product Value ($)
Volatility Scale (1.25x to 0.75x Initial IV)