USD Drop-Back Certificate

S&P 500 Index

Credit Suisse

By Group 06

- Chopra Dhruv
- George Rahul
- Moll-Elsborg Anne

16 November 2023

Table of contents

01

Our Product

Product Description

"USD Drop-Back Certificate" is tied to the S&P 500® Index, with a term of 3 years. It is issued by Credit Suisse AG, London Branch, and allows participation in the performance of the S&P 500® Index. The certificate also has a fixed interest component of **9.85%** p.a. daily accrued and paid at maturity.

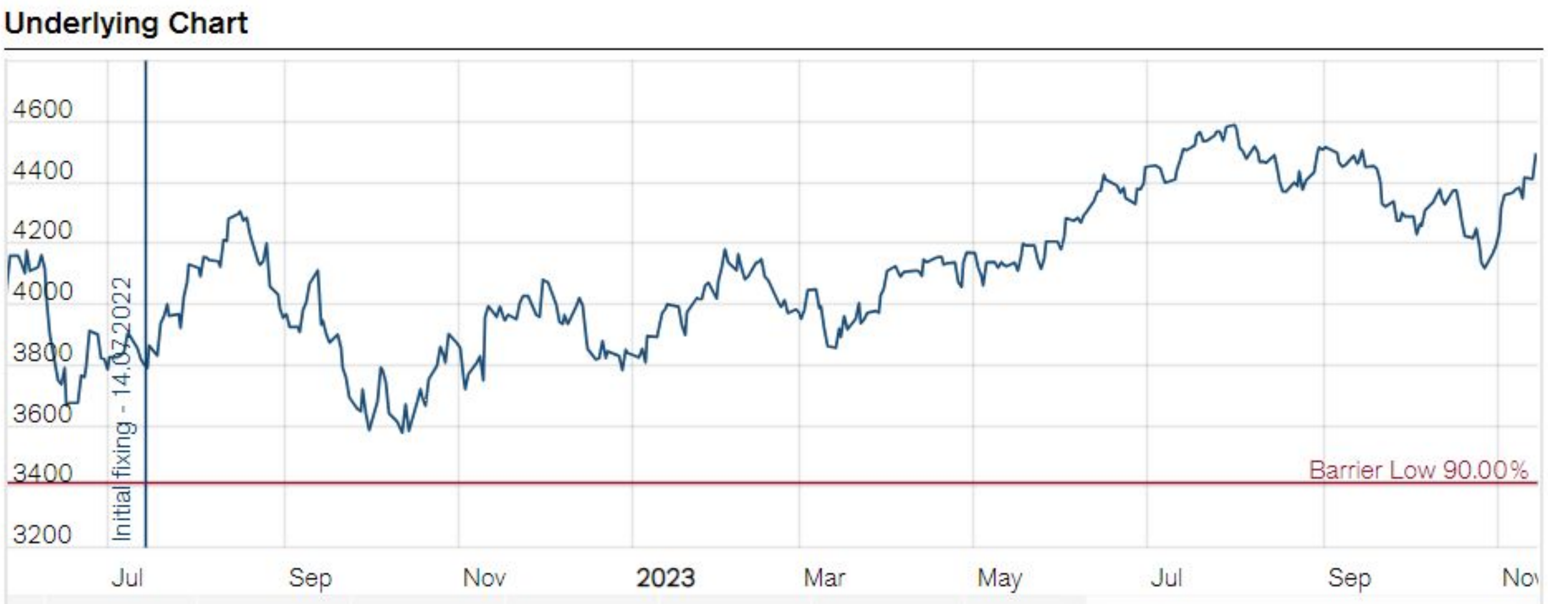
Trigger Events

It has a mechanism of trigger events, where additional investments are made at set index levels if the index drops below certain percentages from its initial level.

Initial fixing date	14 July 2022
Final fixing date	14 July 2025

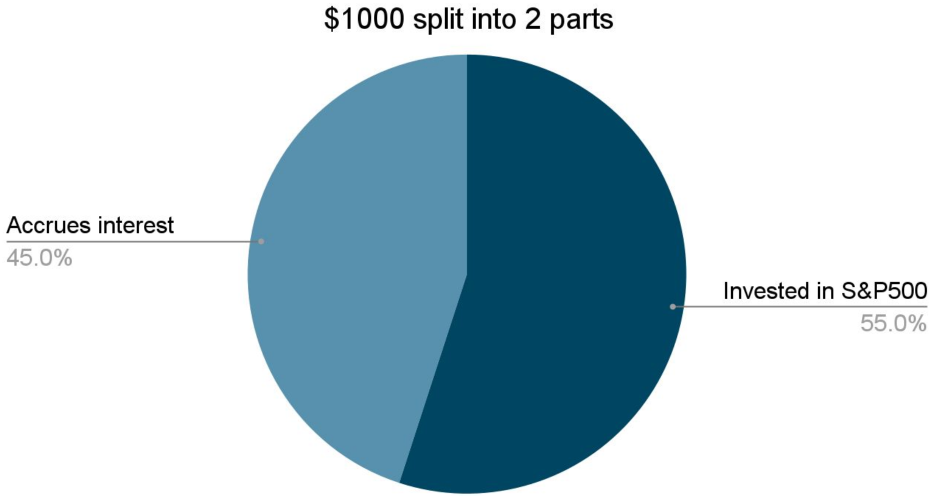
The Underlying Asset S&P 500® Index

The S&P 500 index is a basket of 500 large US stocks, weighted by market cap, and is the most widely followed index representing the US stock market.



The Payoff Function (Simplified)

First let us understand it without any trigger events (barriers).




```python
def calculate_payoff_without_triggers(S_path, Y_0 = 1000, \
                                      rate = 0.0985, years = 3):
    '''
    Computes the payoff of a simulated path without
    the trigger events (barriers).
    '''
    S_0 = 3790.38    # From factsheet
    S_T = S_path[-1]

    #Payoff = Returns from Index + Return from Interest
    Y_T = (0.55*Y_0) * (S_T/S_0) + (0.30*Y_0) * (1 + rate * years)

    return Y_T
```

The Payoff Function (Triggers)

Initial Level	1. Trigger Level 90%	2. Trigger Level 85%	3. Trigger Level 80%
3,790.38	3,411.3420	3,221.8230	3,032.3040

After each trigger event, $150 from the interest portion is invested into the index.

Before Trigger 1:

$550 initially invested and $450 accruing interest.

After Trigger 1:

$550 initially invested, $150 newly invested and $300 accruing interest.

```
def find_trigger_price(S_path, trigger):
    '''
    Returns the price that triggered the event,
    if it exists, else None.
    '''
    # Return the first price that triggers the event
    for price in S_path:
        if price <= trigger:
            return price

    return None
```

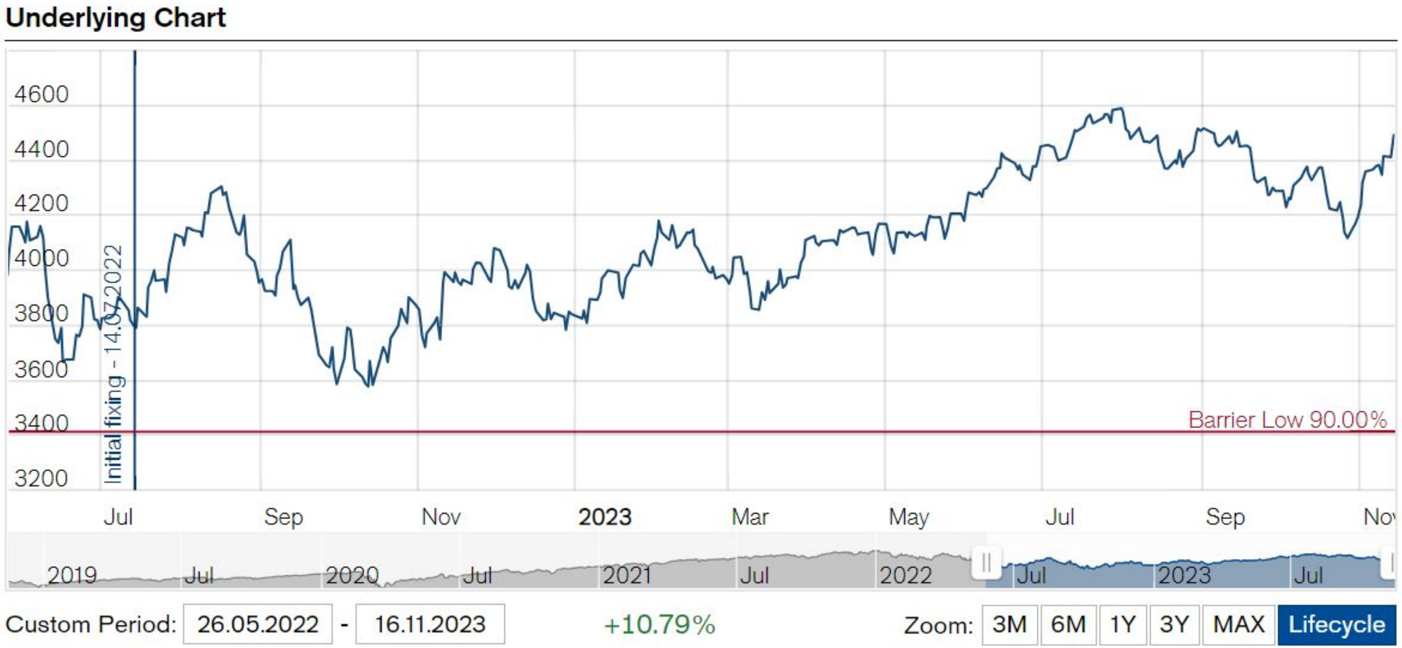
The Payoff Function (Implementation)

```python
def calculate_payoff(S_path, Y_0 = 1000, rate = 0.0985, years = 3):
    '''
    Computes the payoff of a simulated path.
    '''
    S_0 = 3790.38   # From factsheet
    S_t1 = find_trigger_price(S_path, trigger = 3411.3420)   # 0.90 * S_0
    S_t2 = find_trigger_price(S_path, trigger = 3221.8230)   # 0.85 * S_0
    S_t3 = find_trigger_price(S_path, trigger = 3032.3040)   # 0.80 * S_0
    S_T = S_path[-1]

    # 1. Fixed returns independent of trigger events
    Y_T = (0.55*Y_0) * (S_T/S_0)

    # 2. Adjusted returns after trigger events
    if S_t3 is not None:
        multiplier = (S_T/S_t1 + S_T/S_t2 + S_T/S_t3)
    elif S_t2 is not None:
        multiplier = (S_T/S_t1 + S_T/S_t2)
    elif S_t1 is not None:
        multiplier = (S_T/S_t1)
    else:
        multiplier = 0.0

    Y_T += (0.15*Y_0) * multiplier
```

```python
# 3. Daily accrued interest of 9.85% p.a.
    if S_t3 is not None:
        principal = 0.0
    elif S_t2 is not None:
        principal = (0.15*Y_0)
    elif S_t1 is not None:
        principal = (0.30*Y_0)
    else:
        principal = (0.45*Y_0)

    Y_T += principal * (1 + rate * years)

    # Sanity Check
    assert Y_T >= 0.0
    return Y_T
```

Validity of Payoff Function

Although our backtesting does not start from the initial fixing date, the barrier was not hit previously.



02

Black Scholes

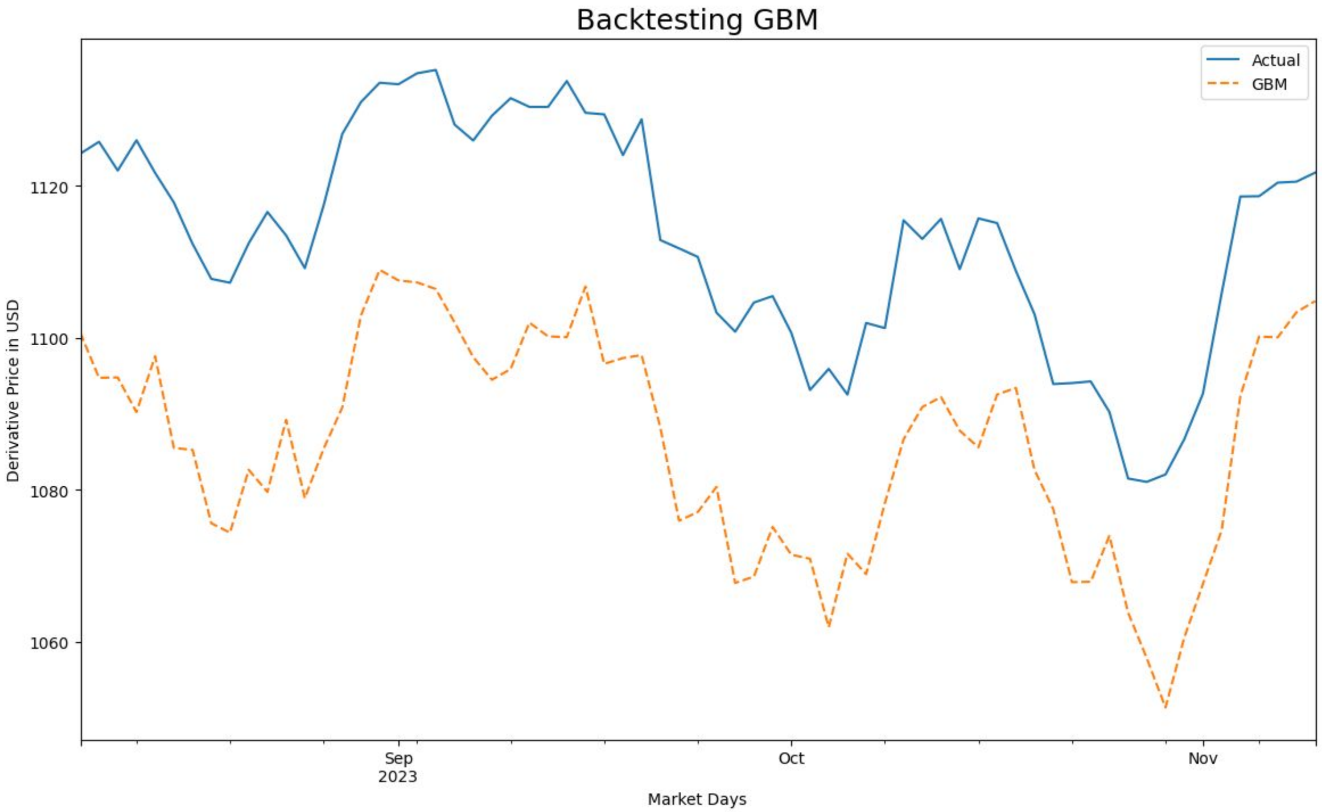
Simulation Settings

1. We priced our product from 9th August to 9th November, 2023.
2. Our backtest window was 252 days long.
3. We ran 20,000 simulations for pricing each day.
4. Simulations were sped up using *Numba*.
5. We interpolated the interest rates using the *Nelson Siegel Svensson* model.

$$r(T) = \beta_0 + \beta_1 \left[\frac{1 - exp(-T/\lambda_0)}{T/\lambda_0}\right] + \beta_2 \left[\frac{1 - exp(-T/\lambda_0)}{T/\lambda_0} - exp(-T/\lambda_0)\right] + \beta_3 \left[\frac{1 - exp(-T/\lambda_1)}{T/\lambda_1} - exp(-T/\lambda_1)\right]$$

NOTE: These settings were used for all subsequent experiments.

Backtesting GBM



Wait, isn't SPX a basket of stocks?

Let $D(t)$ be the value of any dividends paid over $[0, t]$ and interest earned on those dividends. Suppose the asset pays a *continuous dividend yield* of q, meaning that it pays dividends at rate $qS(t)$ at time t. Then D grows at rate

$$\frac{dD(t)}{dt} = \underbrace{qS(t)}_{\textit{Influx of new dividends}} + \underbrace{rD(t)}_{\textit{Interest earned on accumulated dividends}} .$$

If $S \sim GBM(\mu, \sigma^2)$, then the drift in $(S(t) + D(t))$ is

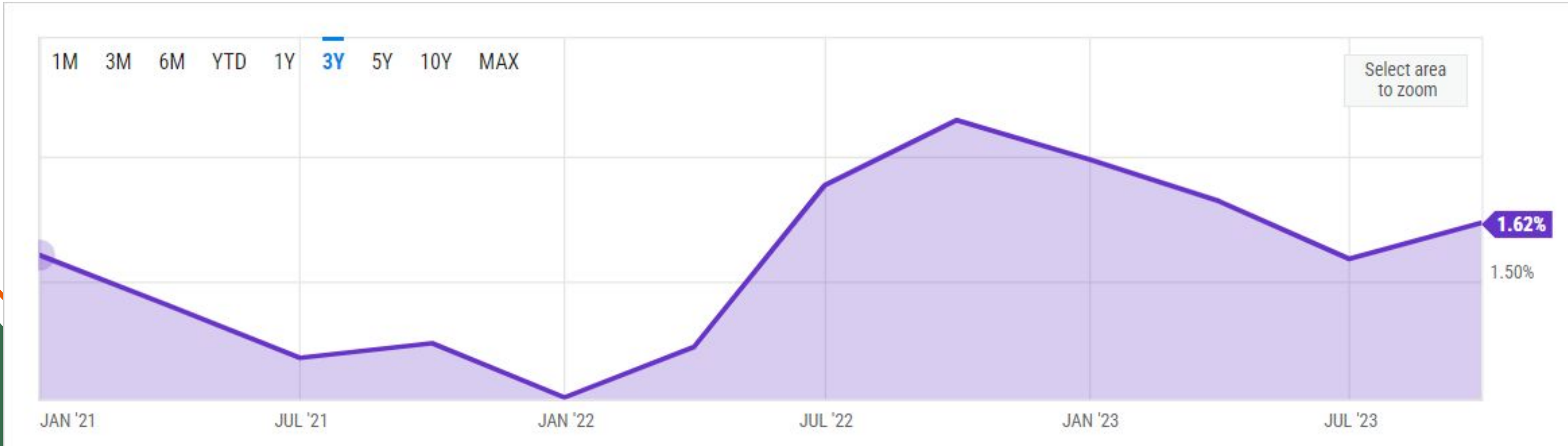
$$\mu S(t) + qS(t) + rD(t).$$

The martingale property, applied to $(S(t) + D(t))$, requires that this drift equal $r(S(t) + D(t))$. We must therefore have

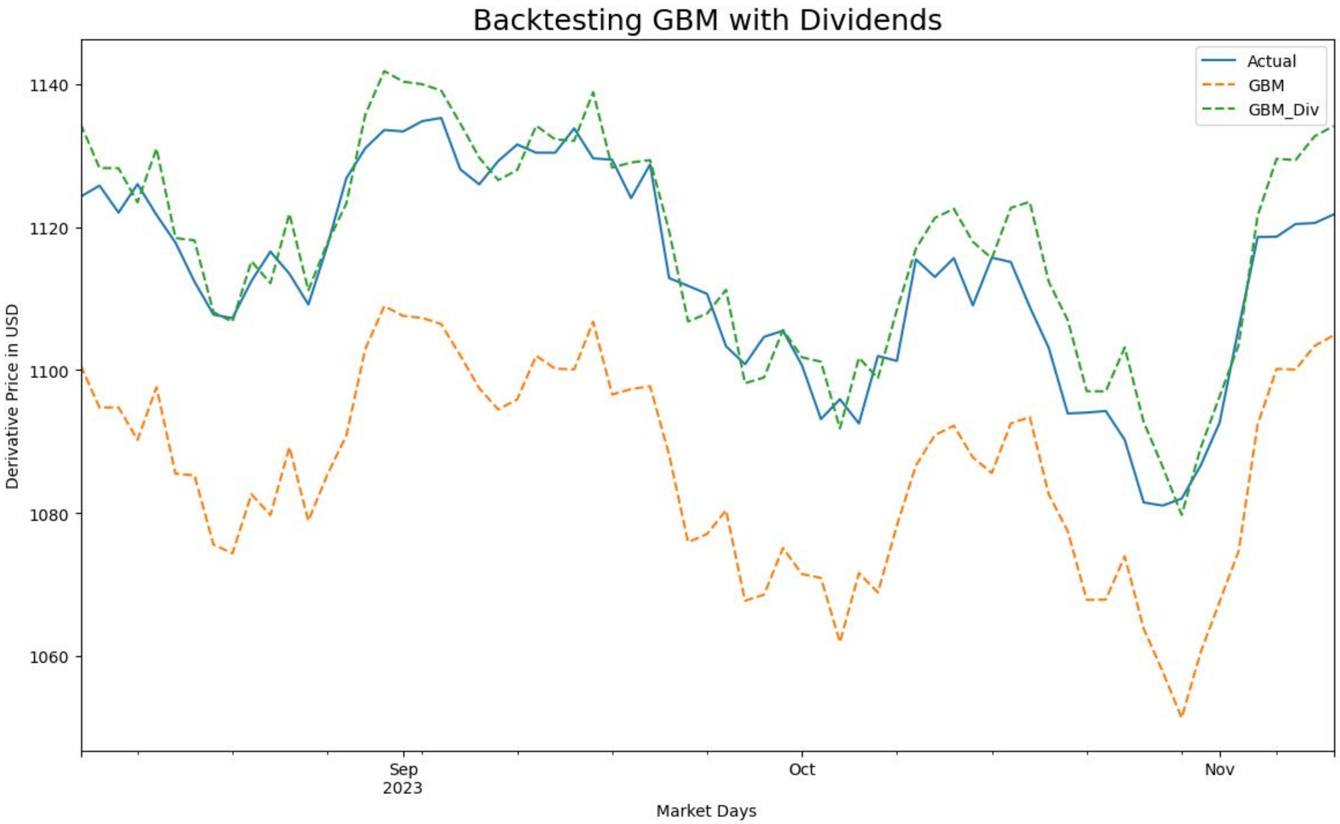
$$\mu = r - q.$$

Enter the S&P 500 Dividend Yield

$$\text{Dividend Yield} = \left(\frac{Sum of Annual Dividends}{Current Market Price of SP500} \right) \times 100$$



Backtesting GBM with Dividends



Error Analysis

We used the mean absolute and mean squared error between predicted and actual prices (for this and all subsequent experiments).

```
Mean Absolute Error for standard GBM:         27.1916
Mean Squared Error for standard GBM:          770.7400

Mean Absolute Error for GBM with dividends:   5.2958
Mean Squared Error for GBM with dividends:    41.5403
```

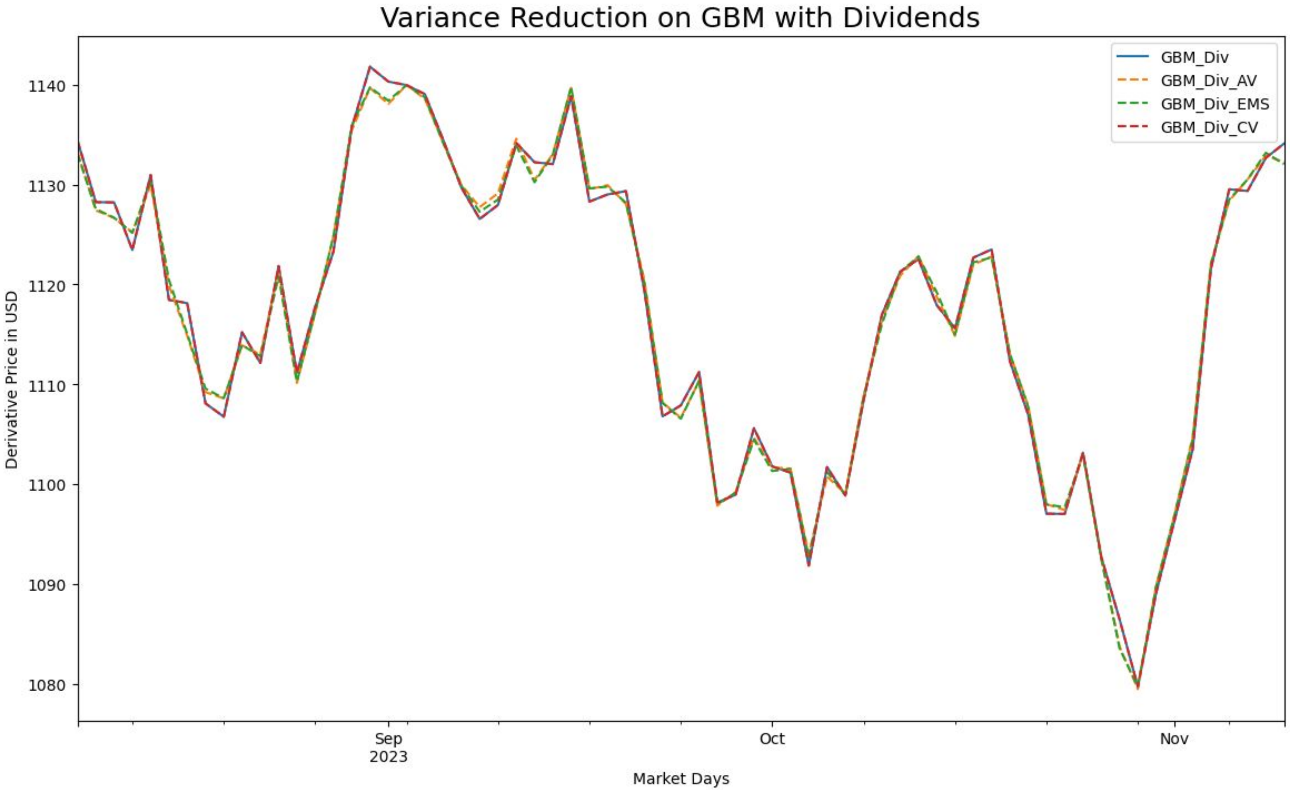
Variance Reduction

We experimented with the following variance reduction methods,

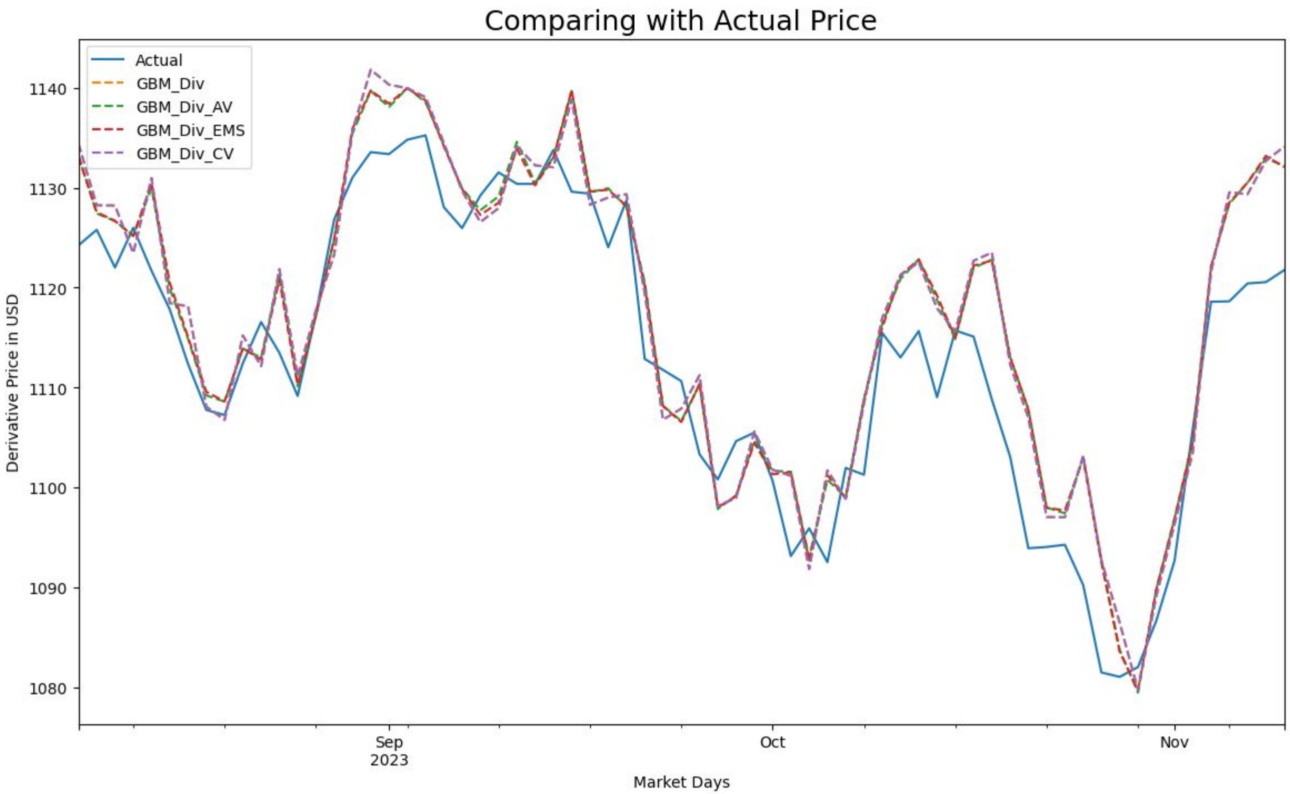
1. Antithetic Variates
2. Control Variates
3. Empirical Martingale Simulation

NOTE: We used the adjusted interest rate accounting for the quarterly dividend rates.

Variance Reduction (Results)



Comparing with Actual Price



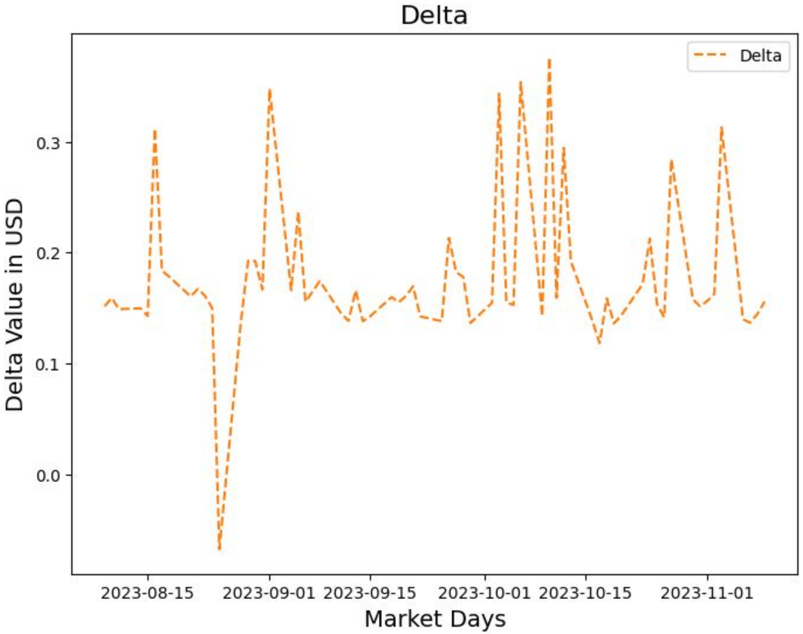
Error Analysis

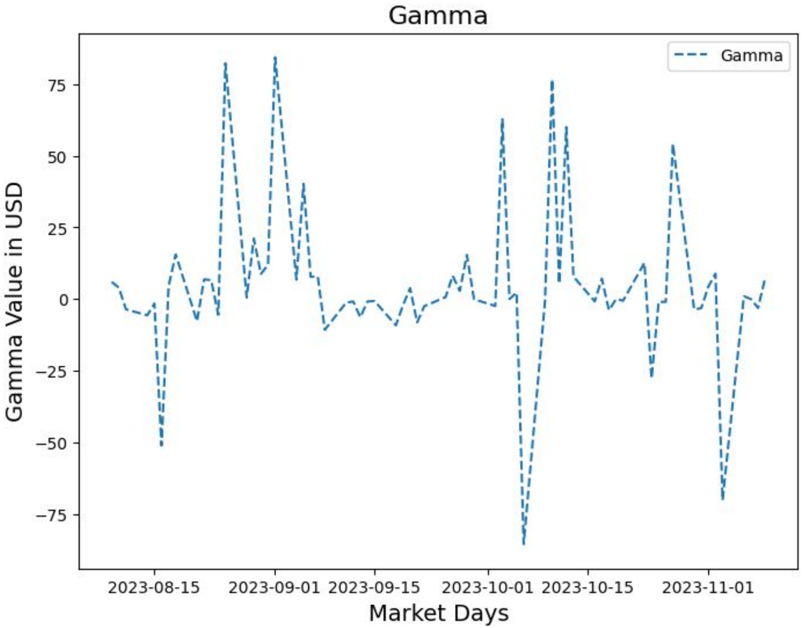
Hard to see from graphs, but there was a slight improvement.

```
Mean Absolute Error for GBM with dividends:     5.2958
Mean Squared Error for GBM with dividends:      41.5403

Mean Absolute Error for GBM_Div with AV:        4.9607
Mean Squared Error for GBM_Div with AV:         38.3676

Mean Absolute Error for GBM_Div with EMS:       5.0107
Mean Squared Error for GBM_Div with EMS:        38.7846

Mean Absolute Error for GBM_Div with CV:        5.2958
Mean Squared Error for GBM_Div with CV:         41.5405
```

Sensitivity Analysis

We simulated the Greeks using the Finite Difference method with h = 0.001 * price





03

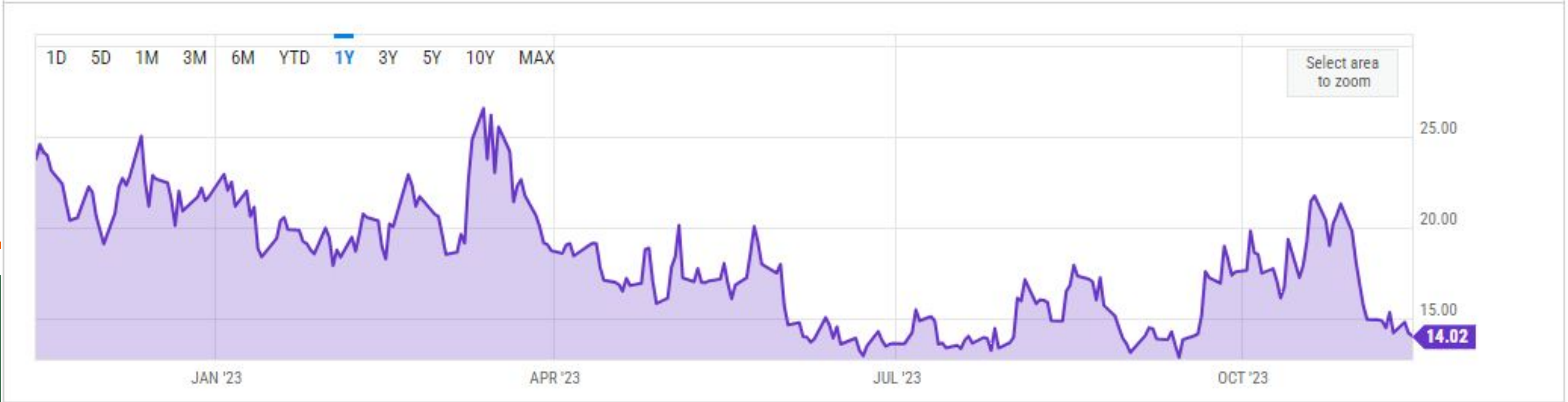
Implied Volatility

Motivation

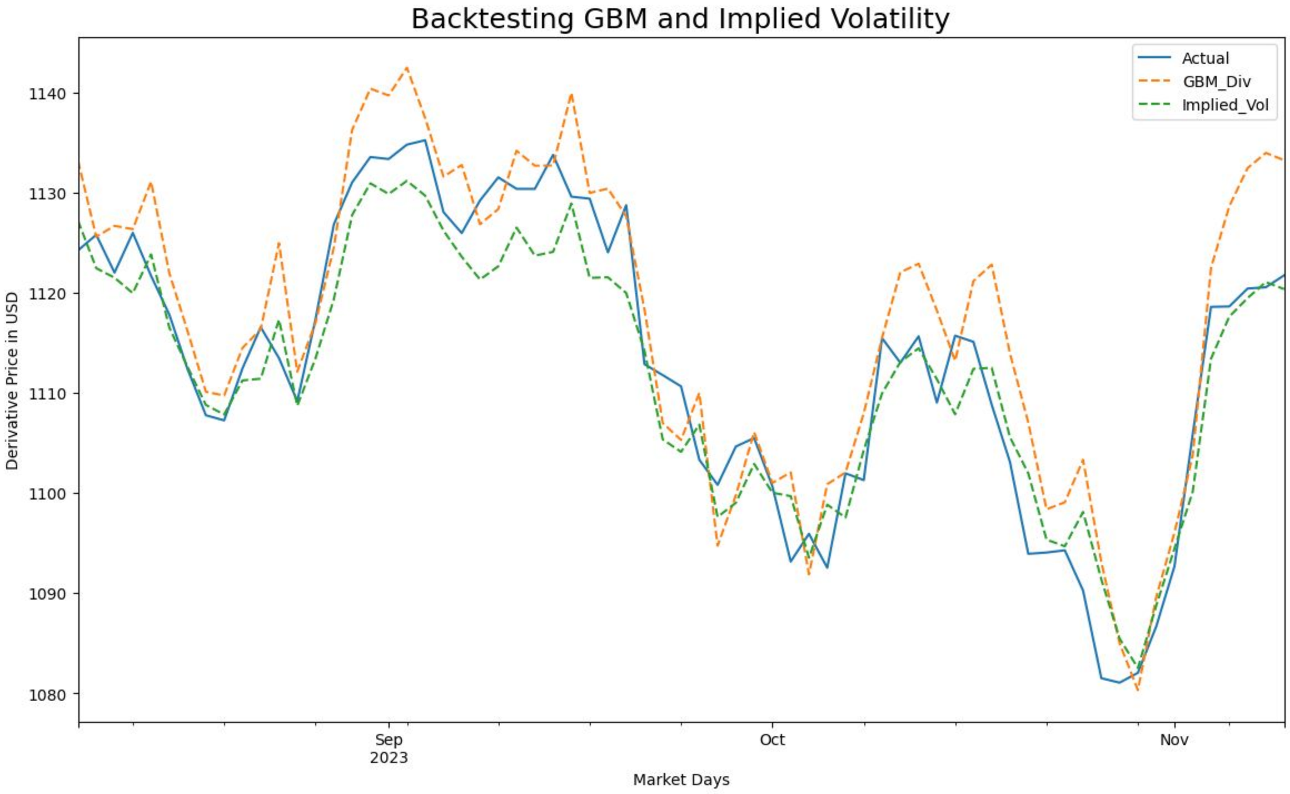
- In Black-Scholes we estimated **σ** using the historical prices, a.k.a., the **historical volatility** approach.
- This may not be the best indicator of the future volatility of the market.
- The **implied volatility** approach considers a **forward-looking** estimate, that is calculated from live option prices.

Enter VIX (by CBOE)

VIX is the ticker symbol and the popular name for the **Chicago Board Options Exchange's** CBOE Volatility Index, a popular measure of the stock market's expectation of volatility based on **S&P 500** index options.



Backtesting Implied Volatility

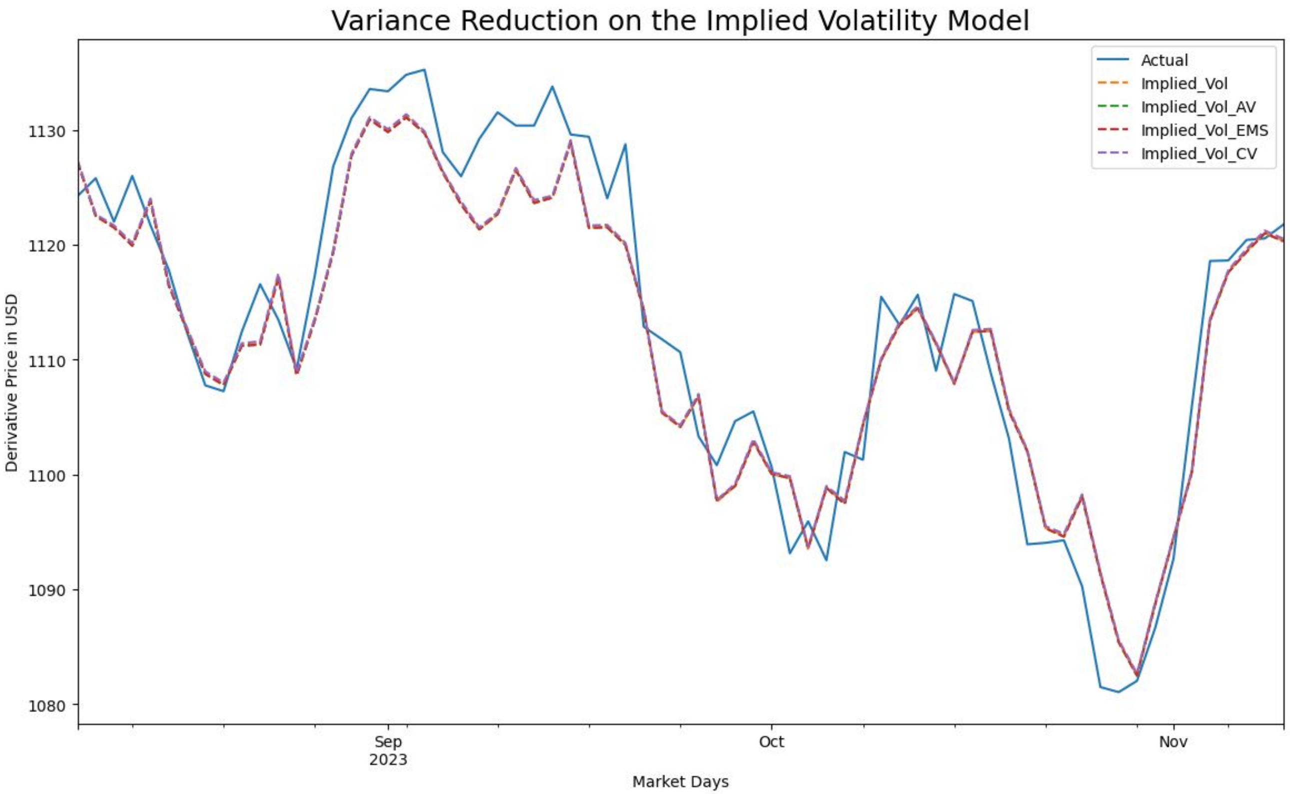


Error Analysis

It had better predictions than the Black-Scholes Model.

```
Mean Absolute Error for GBM with div:          5.3526
Mean Squared Error for GBM with div:           43.5266

Mean Absolute Error for Implied Vol:           3.7179
Mean Squared Error for Implied Vol:            21.0213
```

Variance Reduction



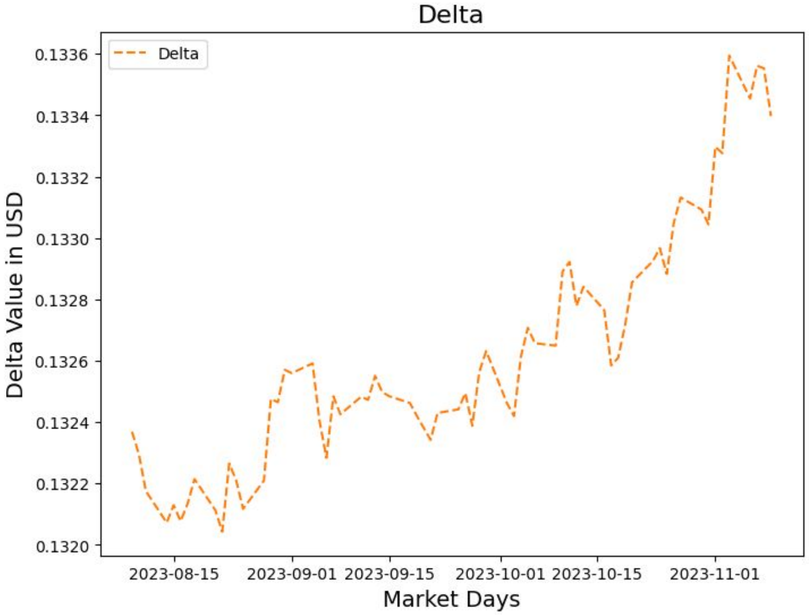
Error Analysis

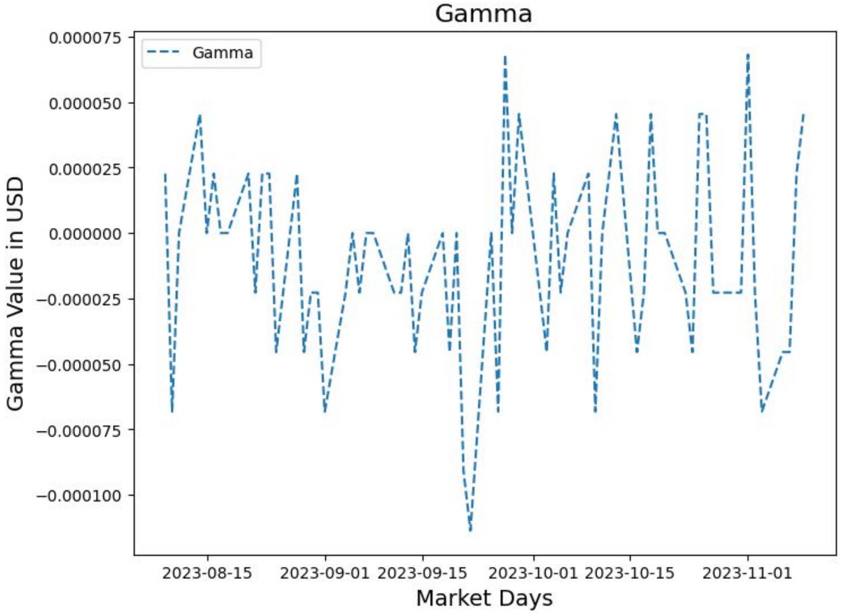
Hard to see from graphs, but there was a very slight improvement.

```
Mean Absolute Error for Implied Vol:            3.7179
Mean Squared Error for Implied Vol:             21.0213

Mean Absolute Error for Implied Vol with AV:    3.7135
Mean Squared Error for Implied Vol with AV:     20.9723

Mean Absolute Error for Implied Vol with EMS:   3.7137
Mean Squared Error for Implied Vol with EMS:    20.9736

Mean Absolute Error for Implied Vol with CV:    3.6700
Mean Squared Error for Implied Vol with CV:     20.5215
```

Sensitivity Analysis

We simulated the Greeks using the Finite Difference method with h = 0.001 * price





04

CIR Model

Model Dynamics

$$dr(t) = \alpha(b - r(t))\,dt + \sigma\sqrt{r(t)}\,dW(t),$$

Euler Discretisation

$$r(t_{i+1}) = r(t_i) + \alpha(b - r(t_i))[t_{i+1} - t_i] + \sigma\sqrt{r(t_i)^+}\sqrt{t_{i+1} - t_i}Z_{i+1},$$

Parameter Estimation

Minimise: $$\frac{(r_t - \Phi r_{t-1})^2}{r_t + \gamma}$$

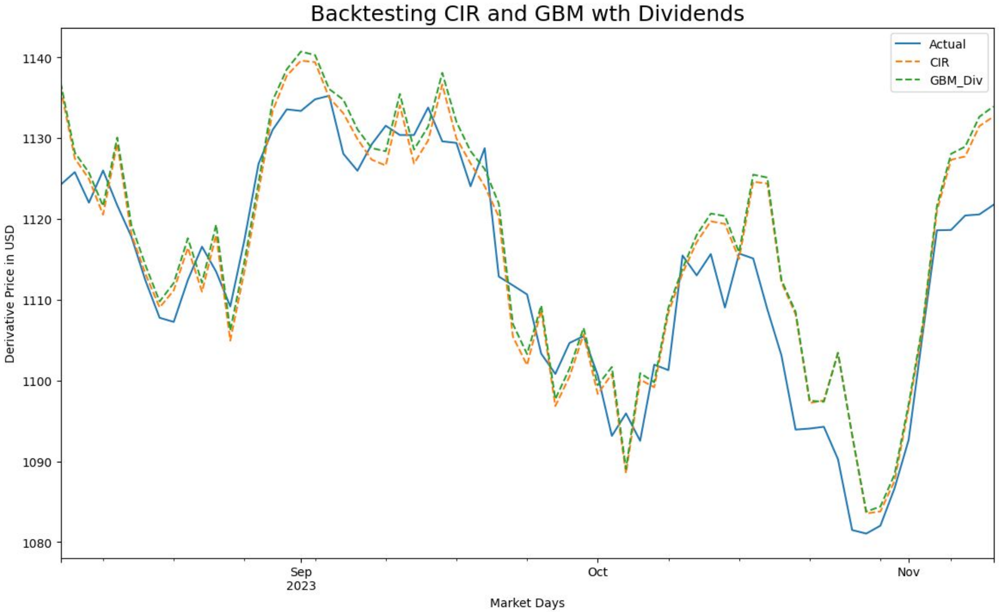
Where,

1. **r_t** = short rate at t - long run average
2. **Phi** = Discrete Drift
3. **Gamma** = long run average

4. **Continuous drift**, κ = -ln Φ = ln(1/ Φ)
5. **Continuous volatility**, σ = √[(2κ*σ^2)/(1-e^(-2κ))]

We use **Continuous drift, Continuous volatility & Long run average** in our simulations

Backtesting CIR




```
Mean Absolute Error for GBM_Div:        5.407055586108778
Mean Squared Error for GBM_Div:         43.38799028539089

Mean Absolute Error for CIR:            5.1283271963728065
Mean Squared Error for CIR:             38.95355947390155
```

Implementation Details

- We use *scipy* optimise library for finding the value of phi that minimises the equation on the previous page
- Need to check that $r < 0$!
- We simulate 2 year bond prices and 3 year bond prices and take their linear interpolation at each window

05

Heston Model

Model Dynamics

$$\frac{dS(t)}{S(t)} = \mu\, dt + \sqrt{V(t)}\, dW_1(t)$$

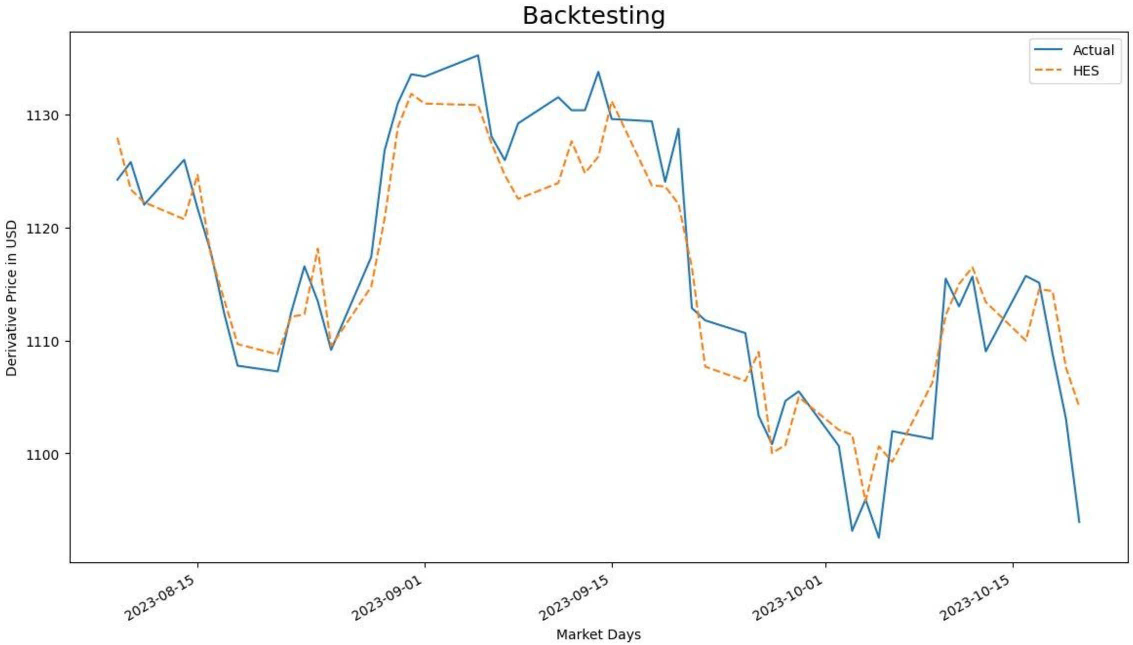
$$dV(t) = \alpha(b - V(t))\, dt + \sigma\sqrt{V(t)}\, dW_2(t),$$

Parameter Estimation

$$\ell(r,k,\theta,\sigma,\rho) = \sum_{t=1}^{n}\Bigg(-\log(2\pi) - \log(\sigma) - \log(V_t) - \frac{1}{2}\log(1-\rho^2) - \frac{(Q_{t+1}-1-r)^2}{2V_t(1-\rho^2)} + \frac{\rho(Q_{t+1}-1-r)(V_{t+1}-V_t-\theta k+kV_t)}{V_t\sigma(1-\rho^2)} - \frac{(V_{t+1}-V_t-\theta k+kV_t)^2}{2\sigma^2 V_t(1-\rho^2)}\Bigg).$$

Minimise the above sum of log-likelihood estimate for each window by varying the five parameters

Results

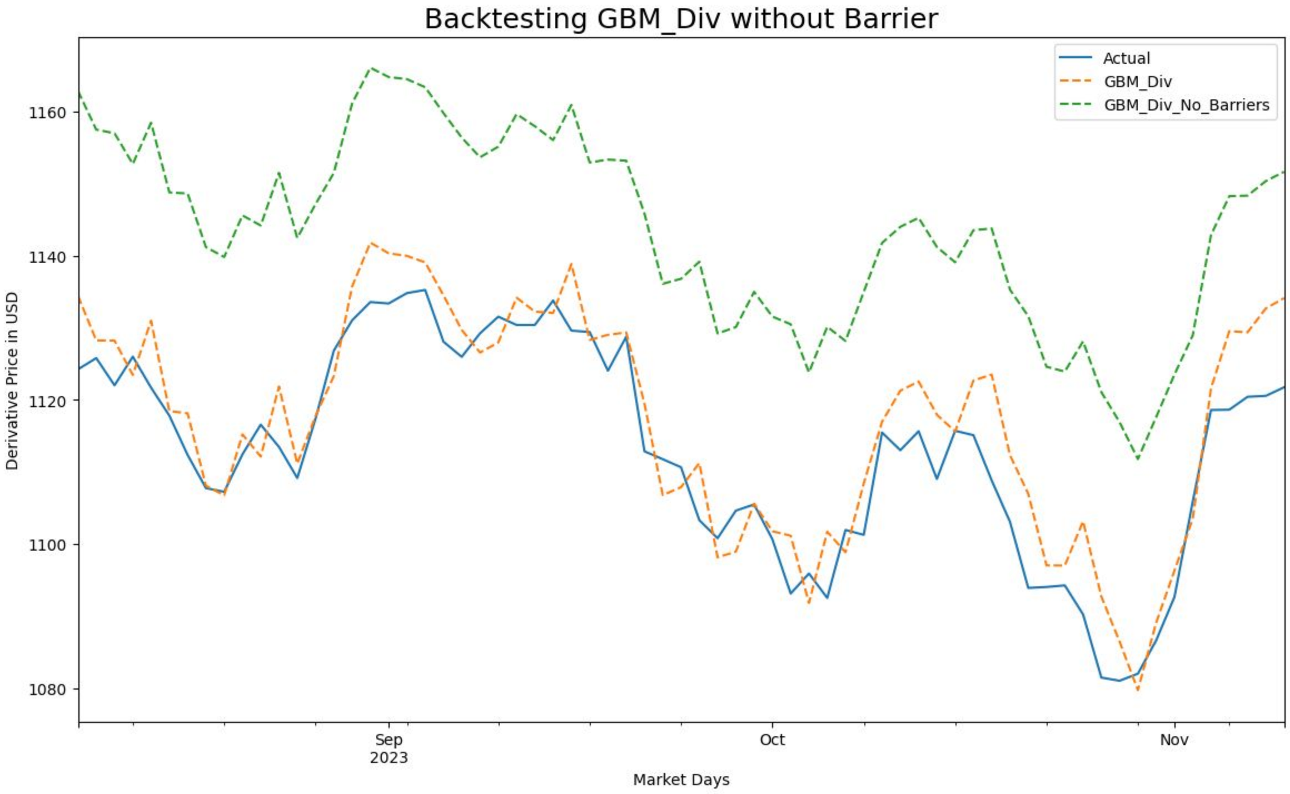



```
Mean Absolute Error for HES:        3.4706635338952676
Mean Squared Error for HES:         18.25832116410608
```

06

Conclusion

Backtesting GBM without Barrier



Our Opinions about the Product

- Option without the Barrier - an investment of $550 in S&P 500 and $450 at 9.85% has **higher** value than the product with the barrier (~$26 higher).
- S&P 500 Year-On-Year return is on average about 6.43%.

Thank you!