MH4518 Simulation Techniques in Finance
2023-24 Semester 1

Project Report
Outperformance Bonus Certificate on EURO STOXX 50

Group 7

Hrishikesh Harish Pai	U2020690L
Sa Ziheng	U2022717L
Yap Guo Dong Nicholas	U2020269C

1. Introduction [Hrishi: 33%, Nicholas: 33%, Ziheng: 33%]

1.1 Product Information

The Outperformance Bonus Certificate for EURO STOXX 50 is a EUR denominated certificate whose underlying asset is tied to the EURO STOXX 50 index that covers the performance of the 50 largest companies in 11 Eurozone countries. This product possesses a Down and Out barrier option, whose threshold is set to 50% of the price at initial fixing.

This product is listed at EUR 1000. It has a lifespan of 36 months, with initial fixing date set to 12 July 2022, at level of EUR 3487.05, final fixing date at 14 July 2025, and the 50% barrier at EUR 1743.53.

1.2 Payoff Function

We simplify the payoff of our product to the following equation:

$$\chi(S_T) = \begin{cases} \max\left(1000\left(1 + 1.5\left(\frac{S_T - S_0}{S_0}\right)\right), 1000\right), & \min(S_t) > V \\ 1000\left(\frac{S_t}{S_0}\right) & \min(S_t) \leq V \end{cases}$$

where χ is the payoff function, S_T is the final asset price, S_0 is the initial asset price, S_t is the asset price at time t, and V is the barrier price. Assuming the index does not reach the 50% barrier, a 150% positive return of the index is obtained.

1.3 Timeline

The timeline of our product is visualised as follows:

In our project, we are interested in pricing our product for every trading day starting from 9 August 2023 to 9 November 2023. We introduce our pricing methodology in the next section.

2. Methodology [Hrishi: 20%, Nicholas: 40%, Ziheng: 40%]

2.1 Sourcing our Data

We summarise our data sources below:

- Stock Data S_t: ^STOXX50E from Yahoo Finance
  - Model parameter: σ (volatility of stock for Geometric Brownian Motion model)
  - Dates: 2 Aug 2022 - 13 Nov 2023 (daily)
- Interest Rate r: Euro Area Yield Curves with 1Y Maturity from ECB
  - Model parameter: r (risk-free interest rate, used in Geometric Brownian Motion and Heston model)
  - Dates: 6 Sep 2004 - 13 Nov 2023 (daily)
- Actual Derivative Price: Web-scraped from Credit Suisse
  - For comparison against our estimated derivative prices
  - Dates: 15 Jul 2022 - 10 Nov 2023 (daily)
- Options Quote Prices for Underlying Asset: Taken from Bloomberg Terminal
  - Model parameters: κ, θ, ρ, ξ, V_0 (Heston model parameters)
  - Dates: 11 Aug 2023 - 15 Mar 2024 (option prices for 11 different expiration dates)

2.2 Simulation Techniques and Variance Reduction

2.2.1 Standard Monte Carlo

The product price will be calculated under the standard Monte Carlo process, where the payoffs for each simulated path is averaged, while they are discounted to the present timestep.

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} X_i$$

2.2.2 Antithetic Variates (Variance Reduction)

Antithetic variates were used as a means to reduce the variance of the simulated price path. For the price paths generated in 2.2.1, taking pairs of random variables X1 and X2 from the standard normal distribution, given that one path is antithetic to the other, the negative covariance of the movements of X1 and X2 would allow for a reduction in the variance of the estimator..

2.2.3 Control Variates (Variance Reduction)

Introducing a control variate that has a known expected value and a nonzero covariance between the known variable Y and the estimator for the variable of interest X, the variance of the estimator can be reduced if X and Y are positively correlated, as this will allow the control variate Y to efficiently correct for the variability of the target variable X

2.2.4 Stratified Sampling

Stratified sampling divides the sample space into strata and obtains independent samples from each stratum, to be averaged. However, due to the path-dependent nature of our product, along with the presence of the 50% barrier, the payoffs could be discontinuous depending on the performance of the option, and a single shot may not be representative of the various price paths and their interactions with the barrier, leading to biased estimates.

2.2.5 Empirical Martingale Correction (Variance Reduction)

To allow for correction of discretization errors, Empirical Martingale Simulation introduces a correction term that accounts for the discrepancy between the true expectation of Monte Carlo simulations and their empirical estimates.

2.3 Our Pricing Process

We summarise our pricing process for each of our models below:

1. Model Parameter Estimation
    a. Risk-free interest rate *r*: For risk-neutral valuation, we use the Euro Area Yield Curves with a 1 year maturity from the European Central Bank to discount payoffs of simulated price paths for the current date of interest.
    b. Starting asset price S_0: We use the Yahoo Finance stock data to obtain the starting asset price for our simulations.
    c. Total simulation duration *T*: For each day in the sliding window, we calculate the total simulation duration *T* from the current date until the final fixing date.
    d. Volatility σ: Using a variable lookback period of M days, we calculate the sample variance of the past M trading days as our volatility, to be used in the Geometric Brownian Motion model.
2. Price Path Simulation
    a. Using the model parameters obtained from the previous stage, we simulate N price paths of the SX5E stock.
3. Payoff Calculation
    a. Next, using the payoff function defined above, we calculate the payoff of each of the N price paths.
4. Derivative Price Estimation

a. Finally, we estimate the payoff of our derivative for our date of interest by discounting the final payoff using our risk-free interest rate r and the number of timesteps from current time t to the final fixing date T.

3. Modelling of Underlying Asset

3.1 Geometric Brownian Motion Model [Hrishi: 20%, Nicholas: 60%, Ziheng: 20%]

3.1.1 Model Description

The Geometric Brownian Motion (GBM) model is a mathematical model commonly used to describe the stochastic movement of asset prices over time.

The GBM model is expressed as a stochastic differential equation:

$$dS_t \; = \; \mu S_t dt + \sigma S_t dW_t$$

where dS_t is a small change in asset price, μ is the average return rate, σ is the volatility of the asset, and dW_t is a small increment in a Wiener process.

Under the risk-neutral measure $\mathbb{Q}$, we replace the average return rate μ with the risk-free interest rate r as shown below:

$$dS_t = rS_t dt + \sigma S_t dW_t$$

We can then use Ito's lemma to simulate our asset price paths using the below formula:

$$S_j^{(i)} = S_{j-1}^{(i)} \exp\left(\left(r - \frac{\sigma^2}{2}\right)\Delta t + \sigma\sqrt{\Delta t}Z^{(i)}\right)$$

3.1.2 Simulation Setup

For our simulation, we run 1,000,000 simulations for every day in the sliding window, from 9 August 2023 to 9 November 2023 for a total of 66 trading days, with a lookback period of 252 trading days (1 year). We repeat this simulation for lookback periods of 21 trading days (1 month) and 63 trading days (3 months) as an experiment to evaluate whether a shorter lookback period would be more effective in pricing our derivative.

For calculation/estimation of model parameters from historical data, we take the following values for every trading day in the sliding window(e.g. 9 Aug 2023):

- r: Taken from current day (e.g. 9 Aug 2023)
- S_0: Taken from current day (e.g. 9 Aug 2023)
- σ: Taken from previous M timesteps (e.g. 252 trading days)
- T: Calculated from current day (e.g. 485 working days from final fixing date)

3.1.3 Estimating Sensitivities

We estimate delta and gamma sensitivities using the finite-difference method. We estimate delta using the following formula:

$$\delta \approx \frac{f(S+h) - f(S-h)}{2h}$$

Gamma is estimated using the following formula:

$$\Gamma \approx \frac{f(S+h) - 2f(S) + f(S-h)}{h^2}$$

For our project, we limit calculation of the above Greeks to only the Monte Carlo method with 10,000 simulations, as sensitivity estimation is very memory-intensive and requires simulation of asset price paths 2 additional times.

3.2 Heston Model [Hrishi: 20%, Nicholas: 20%, Ziheng: 60%]

3.2.1 Model Description

The Heston model, which is an extension of the widely recognized Black-Scholes model for options pricing, is characterised by a set of Stochastic Differential Equations (SDEs). These equations are used to model the behaviour of asset prices, wherein both the price of an asset and its volatility are subject to random, Brownian motion-like movements.

Under the risk-neutral measure $\mathbb{Q}$, the Heston Model SDEs are given by:

$$\begin{aligned} dS(t)/S(t) &= r dt + \sqrt{V(t)} dW_S^{\mathbb{Q}}(t), \\ dV(t) &= [\kappa(\theta - V(t)) - \rho(\mu - r)\xi] dt + \xi \sqrt{V(t)} dW_V^{\mathbb{Q}}(t) \end{aligned}$$

where κ is the mean-reverting rate of variance, θ is long-run average variance, ξ is the volatility of the variance process, and ϱ is the correlation between the stock price and variance process.

Under the Heston model, the European call option price at time t is given as follows:

$$C = S_t \cdot \left[\frac{1}{2} + \frac{1}{\pi}\int_0^{\inf} Re\left[\frac{e^{-is\ln K} f_1(s,\nu,x)}{is}\right] ds\right] - Ke^{-rt}\left[\frac{1}{2} + \frac{1}{\pi}\int_0^{\inf} Re\left[\frac{e^{-is\ln K} f_2(s,\nu,x)}{is}\right] ds\right]$$

which involves solving 2 integrals. It can be simplified to a single integral form to reduce computational runtime and the formula is given as follows:

$$C = \frac{1}{2}(S_t - Ke^{-r(T-t)}) + \frac{1}{\pi}\int_0^{\infty} Re\left[e^{r(T-t)}\frac{f(s-i)}{is.K^{is}} - K\frac{f(s)}{is.K^{is}}\right] ds \quad .$$

3.2.2 Simulation Setup

For the Heston model, as the size of our data obtained was insufficient for the entire sliding window, we perform pricing for our derivative only for the date of 9 August 2023.

Using the Bloomberg terminal, we collect the European call option quote prices for 11 maturity dates ranging from 11 Aug 2023 to 15 Mar 2024 with different strike prices to construct a volatility surface. We also use the last reported interest rate r from ECB and the last traded price of the underlying to perform parameters (κ, θ, ρ, ξ, V_0) calibration. The objective function defined is:

$$\min_\theta \left(\sum_{i=1}^{N} (P_i(\theta) - M_i)^2 \right)$$

which is optimised by the optimisation algorithm: Levenberg Marquardt.

With the parameters obtained through calibration, we proceed to simulate the underlying price with discrete Heston SDEs for 1,000,000 simulations. We then apply the payoff function and discounting factor to the average simulated underlying price to estimate our option price for 9 Aug 2023. Results are discussed below.

4. Results [Hrishi: 33%, Nicholas: 33%, Ziheng: 33%]

4.1 Historical Data

4.1.1 Geometric Brownian Motion Model

Our parameter σ obtained from historical data for different lookback periods are shown below:

Parameter(s)	Lookback Period	Value (9 Aug 2023)	Value (9 Nov 2023)	Average Value Across Window
σ	21	0.1721991002559662	0.13019900023937225	0.13611777126789093
	63	0.1524956226348877	0.12539911270141602	0.14484648406505585
	252	0.1707075983285904	0.1513691544532776	0.1620490998029709

Table 1: GBM model parameters obtained from historical data

4.1.2 Heston Model

Our parameters obtained from the model calibration process are as follows:

Parameters	Initial Value	Lower Bound	Upper Bound	Results
κ	0.5	10^{-2}	3	**2.99995740**
θ	0.5	10^{-2}	3	**0.01003860**
ρ	-0.5	-1	0	**-0.64633133**
ξ	0.5	10^{-2}	5	**0.01000034**
$\mathbf{V_0}$	0.17	10^{-2}	1	**0.99999979**

Table 2: Heston model parameter values obtained from model calibration

4.2 Estimated Derivative Prices

4.2.1 Geometric Brownian Motion Model

The graph of our estimated derivative prices against the actual derivative prices is shown below:

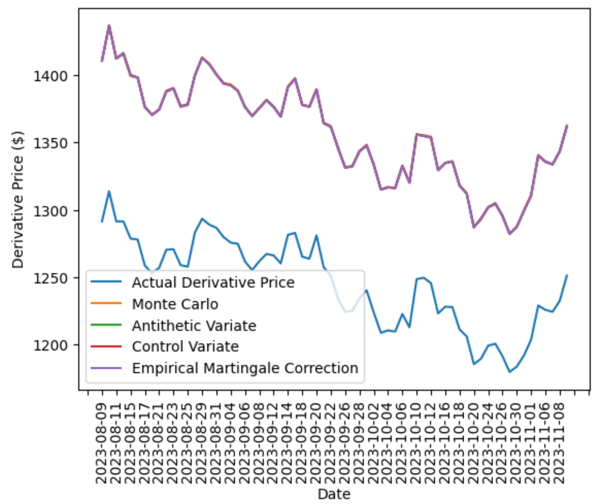

Figure 1: Actual vs Estimated Derivative Prices for lookback 252 days

MEV - mean estimate variance, VRP - variance reduction percentage, MAE - mean absolute error

	Lookback 252 days			Lookback 63 days			Lookback 21 days		
Simulation Technique	**MEV**	**VRP (%)**	**MAE**	**MEV**	**VRP (%)**	**MAE**	**MEV**	**VRP (%)**	**MAE**
Standard Monte Carlo	0.13552	-	111.41	0.11119	-	104.33	0.10240	-	102.74
Antithetic Variates	0.13554	-0.01	111.42	0.11123	-0.03	104.43	0.10242	-0.02	102.74
Control Variates	0.00387	97.14	111.41	0.00253	97.73	104.43	0.00223	97.82	102.75
Empirical Martingale Correction	0.13548	0.03	111.19	0.11117	0.01	104.20	0.10240	-0.00	102.52

Table 3: Metrics for different simulation techniques with different lookback periods with 1,000,000 simulations.

The lookback period of 21 days achieves the lowest mean absolute error of around EUR102 from the actual derivative price, followed by the lookback period of 62 days with mean absolute error of around EUR104 and lookback period of 252 days with mean absolute error of around EUR111. From here, we surmise that the SX5E stock is a volatile stock, where we would benefit from obtaining model parameters using shorter lookback periods.

4.2.2 Heston Model

Simulation Technique	Estimate Price	Estimate Variance	Mean Absolute Error
Standard Monte Carlo	1504.36	1.23866	212.90

Table 4: Estimated price for 9 Aug using Heston Model and Monte Carlo simulation

4.3 Estimated Sensitivities

4.3.1 Geometric Brownian Motion Model

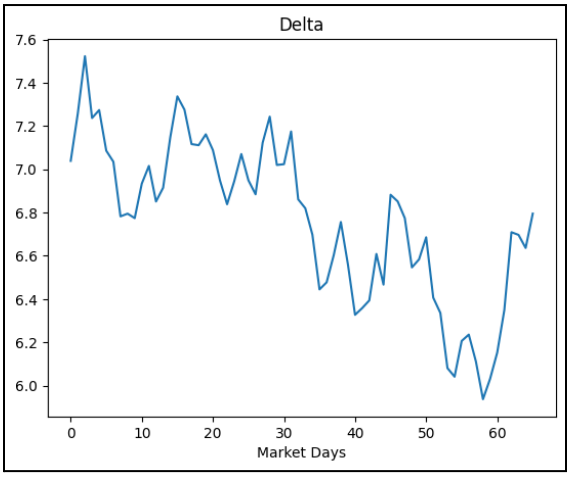

Figure: Estimates of δ

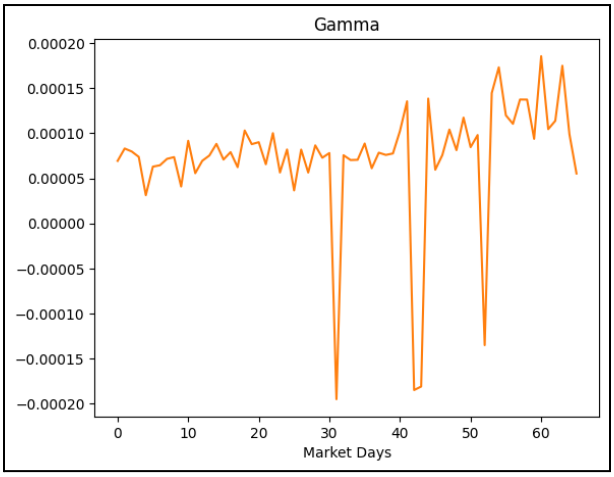

Figure: Estimates of Γ

The delta and gamma sensitivities obtained using the Monte Carlo technique for the GBM model after 10,000 simulations are shown above. The delta graph tells us how a 1 EUR change in underlying asset price affects the derivative price, while the gamma graph tells us how a 1 EUR change in underlying asset price affects the delta for each market day.

4.4 Percentage of Barrier Triggered

4.4.1 Geometric Brownian Motion Model

Below, we compute the mean number of times the barrier was reached out of 1 million price paths across the 66 days in the sliding window, and the mean barrier knock-out percentage.

	Lookback 252 days		**Lookback 63 days**		**Lookback 21 days**	
Simulation Technique	**Mean No. of Barrier Knock-outs**	**Mean Barrier Knock-out %**	**Mean No. of Barrier Knock-outs**	**Mean Barrier Knock-out %**	**Mean No. of Barrier Knock-outs**	**Mean Barrier Knock-out %**
Standard Monte Carlo	29.575758	0.0030%	4.287879	0.0004%	27.984848	0.0028%
Antithetic Variates	29.924242	0.0030%	4.575758	0.0005%	28.045455	0.0028%
Control Variates	29.136364	0.0029%	4.196970	0.0004%	27.257576	0.0027%
Empirical Martingale Correction	29.378788	0.0029%	4.090909	0.0004%	29.333333	0.0029%

Table 5: Mean barrier knock-out statistics for 1,000,000 simulated price paths across 66 days in sliding window

5. Simulation Enhancements [Hrishi: 33%, Nicholas: 33%, Ziheng: 33%]

To decrease the simulation runtime, we implemented 2 major simulation enhancements which allowed us to reduce the time needed for simulation of 1,000,000 simulations over 66 days for 5 different techniques from 30 hours to a bit more than an hour. We utilised the Numba library to compile the simulation functions (MC, AV, CV, SS, EMS) from Python code to machine code, by rewriting the functions to comply with the Numba compiler and adding a @jit function decorator in front of the simulation functions. This improved performance greatly, and effectively sped up simulation by about 4x. We also utilised the concurrent.futures Python library to parallelize simulations across different days in the sliding window. This allowed us to cut simulation runtime by the number of CPUs assigned to the simulation. Since we used 7 CPUs, we cut runtime by about 7x.

6. Conclusion [Hrishi: 33%, Nicholas: 33%, Ziheng: 33%]

In summary, the shape of our estimated derivative prices produced by the GBM model generally align with the shape of the actual derivative prices, capturing significant declines and increases. However, there is a noticeable offset between the simulated prices and the actual prices of about EUR100 that our GBM model was unable to precisely replicate. As the GBM model uses constant variance and constant risk-free rate, this may affect its performance. It also does not consider macroeconomic factors like market movements and other forward-looking indicators. For future improvement, we can consider models which treat both the risk-free interest rate and the volatility as stochastic differentials.

7. References

1. Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The review of financial studies*, 6(2), 327-343.
2. https://calebmigosi.medium.com/build-the-heston-model-from-scratch-part-i-17bde00122a0
3. https://calebmigosi.medium.com/build-the-heston-model-from-scratch-in-python-part-ii-5971b9971cbe
4. https://calebmigosi.medium.com/build-the-heston-model-from-scratch-in-python-part-iii-monte-carlo-pricing-5009c1ba7d29
5. https://github.com/CalebMigosi/code-more/blob/main/EquityOptionsPricing/Heston%20Pricing%201.ipynb
6. Simulation Techniques in Financial Risk Management (2nd Edition) by N. H. Chan and H. Y. Wong (2015)
7. Handbook of Financial Risk Management: Simulations and Case Studies by N. H. Chan and H. Y. Wong (2013)

Appendix

Web scraping of Credit Suisse Derivative

- To obtain the actual derivative price from the Credit Suisse page, our group chose to scrape the prices from the website instead of manually entering the values.
- As of the time of writing, the Credit Suisse derivative page used AmCharts v3.20.17 to present the derivative price as well as the underlying asset price. This data was stored in a JavaScript object in the page called AmCharts, which consisted of two types of chart objects (stock charts and serial charts)
    - Stock charts contained the dataset title, while serial charts contained the data (without the title), so code had to be written to link the two types of charts
- To scrape the data from the website, we used a library called Selenium, which programmatically interacts with websites by opening an actual browser window and simulating real interactions with the elements. This library will need to be installed before using the code. The written code is in a file called “credit_suisse_scraper.py” in our project code, which should be able to perform scraping for other derivatives by updating the variable named “DERIVATIVE_SUBLINK” in the code.
- Upon running the code, every asset in the page will be downloaded as a CSV into the folder specified by the variable DATA_FOLDER. However, there may be instances where the charts load in a different order, which causes certain assets to not be downloaded. In that case, the code should be re-run to download the charts again.
- This code has been tested on and works for our SX5E derivative, as well as the Tesla derivative, and can hopefully aid projects for future batches of MH4518.