AIMS Mathematics, 10(5): 10452–10470.
DOI: 10.3934/math.2025476
Received: 23 December 2024
Revised: 10 April 2025
Accepted: 25 April 2025
Published: 07 May 2025

https://www.aimspress.com/journal/Math

Research article

Pricing five-asset Equity-Linked Securities (ELS) with step-down and knock-in barrier conditions on Android platform

Junseok Kim[1], **Juho Ma**[1], **Hyunho Shin**[1] **and Hyundong Kim**[2,*]

[1] Department of Mathematics, Korea University, Seoul 02841, Republic of Korea
[2] Department of Mathematics and Physics, Gangneung-Wonju National University, Gangneung 25457, Republic of Korea

* **Correspondence:** Email: hdkim@gwnu.ac.kr.

Abstract: In this work, we propose a computational algorithm for calculating the fair price of five-asset Equity-Linked Securities under a step-down structure and a knock-in barrier condition. By using the Android platform, we implement a Monte Carlo simulation to accurately compute the fair price of Equity-Linked Securities with five-asset dynamics, which offers the advantage of location-independent pricing through mobile devices. We use the Cholesky decomposition to generate correlated random numbers based on the correlation matrix of the five underlying assets. To generate the discrete stock price paths, a discretized stochastic differential model is considered under the geometric Brownian motion assumption. The findings confirm the usefulness of the Monte Carlo simulation in pricing five-asset step-down Equity-Linked Securities derivatives and demonstrate the practicality of conducting advanced financial calculations on mobile devices. The implementation of this algorithm on the Android platform represents a significant advancement in the development of multi-asset Equity-Linked Securities, which offer the potential for higher coupon yields. Moreover, this research highlights the potential for further innovations in Fintech and suggests that the integration of technology has the capacity to improve the investment process by improving the accessibility and efficiency of financial tools.

Keywords: option pricing; Equity-Linked Security; multi-asset; Monte Carlo method; Android platform
Mathematics Subject Classification: 39A50, 60H35, 65C30, 65Y04, 65Y20, 91G20, 91G60

1. Introduction

An Equity-Linked Security (ELS) is a financial instrument issued and sold by security companies, whose returns are linked to the performance of individual stock prices or stock indices. These financial

products are risky and there is a possibility of a principal loss. ELS crossed the knock-in barrier, which is tied to the Hang Seng China Enterprises Index (HSCEI), declining by as much as 58% between 2021 and 2024 [1]. This event resulted in significant investor losses. Investors lack effective tools to accurately evaluate the risk of loss inherent in ELS financial products. For these reasons, there is a clear need for a framework that enables investors to comprehensively understand the structural characteristics and risk levels of ELS financial products. The ELS financial products linked to multiple underlying assets typically offer higher coupon rates, but they also entail an increased risk of substantial losses. In [1], the authors investigated the application of Markov Regime-Switching models in pricing ELS, especially in response to significant losses observed in HSCEI-linked ELS products in South Korea during 2024. The motivation stems from the inadequacy of conventional methods (e.g., historical or implied volatilities) to capture long-term market regime changes and tail risk.

A number of studies have been conducted on the pricing of ELS, especially those structured with one to four underlying assets. Fractional derivatives serve as a critical tool in financial mathematics by enabling the modeling of long-range dependence and memory effects observed in asset dynamics. Their incorporation into pricing and volatility models enhances the representation of non-local temporal behaviors, leading to improved analytical accuracy and robustness in complex financial systems. The general time-fractional diffusion equation (TFDE), as addressed in [2], provides a valuable framework for modeling anomalous diffusion and memory-dependent processes. The authors introduce a generalized fractional derivative involving both a scale function and a weight function, thereby allowing the model to more accurately capture complex physical behaviors than standard Caputo-type derivatives. Their aim was to derive analytical solutions, analyze regularity, and develop high-order numerical algorithms for both deterministic and stochastic versions of the TFDEs [3, 4]. In [5], a compact finite difference scheme was presented for solving the time-fractional Black–Scholes model used in European option pricing. This model generalizes the classical Black–Scholes equation by incorporating a Caputo fractional derivative in time to better capture memory and hereditary properties inherent in financial markets. Pricing frameworks [6, 7] have been studied for three-asset step-down ELS options by replacing traditional geometric Brownian motion with fractional Brownian motion. These studies can provide a more realistic representation of the long-memory behavior observed in financial markets, which is often inadequately described by conventional models with independent increments. On the other hand, in the field of ELS pricing, the Monte Carlo method excels at modeling the random behavior of underlying assets, making it well-suited for valuing derivatives linked to multiple assets and those with complex payoff structures [8–10]. Its application allows for the simulation of a vast number of possible scenarios, providing accurate estimates of the expected value of the securities while accounting for the inherent uncertainty and volatility in the financial markets [11]. The Monte Carlo method is particularly advantageous when analytical solutions are not readily available or are computationally intensive, thus offering a versatile and robust approach for ELS pricing in practice. Moreover, the Monte Carlo approach to option pricing has been implemented using Compute Unified Device Architecture (CUDA) [12]. Recently, the authors in [13] explored a quantum computing approach for pricing options, particularly focusing on the use of the amplitude estimation algorithm to achieve a quadratic speed-up over classical Monte Carlo methods. In studies on the structure of multi-asset ELS, the fair value of ELS has been calculated using both the finite difference method [14] and the Monte Carlo method [8–11]. Additionally, the efficiency of the calculations has been enhanced by using the Brownian bridge method [15–17]. These methods were implemented on

the Android platform, enabling investors to independently calculate the fair value. While prior studies have proposed analytical or numerical methods including finite difference schemes and FMB-based models, they often lack implementation in real-time computing environments or mobile platforms. Furthermore, most existing works did not address the practical computational challenges of pricing multi-asset (five or more) ELS with both step-down and knock-in barrier features.

In this study, we propose an analysis of a five-asset structured ELS product, and apply the Monte Carlo simulation technique as a numerical method to calculate both its fair value and the associated risk rate. The Monte Carlo simulation is particularly suited for this task due to its ability to model complex, path-dependent features and capture the stochastic nature of financial markets. To facilitate practical application, this model is implemented on the Android platform, enabling real-time computation and accessibility for end-users. The risk rate, in this context, is defined as the ratio of the number of times the knock-in barrier is breached during the simulation trials to the total number of trials conducted. This risk rate metric functions as a quantitative indicator of the likelihood of a knock-in event occurring, thereby providing a measurable gauge of the risk associated with the ELS financial products. By incorporating this approach, we aim to enhance the understanding of the pricing dynamics and risk profiles of multi-asset-linked ELS products in a market-driven environment.

The contents of this study are organized as follows. In Section 2, we present the numerical algorithm using the Monte Carlo simulation. In Section 3, computational experiments are conducted using the proposed Android platform. Finally, Section 5 presents the conclusion of the findings of this paper.

2. Numerical approach for the simulation

In this section, we describe the numerical methodology used to simulate the fair value of five-asset ELS with step-down and knock-in barrier structures. We use the Monte Carlo method with correlated geometric Brownian motion to model asset dynamics. The correlation among the underlying assets is incorporated through the Cholesky decomposition of the correlation matrix.

Now, we provide a detailed explanation of how the Cholesky decomposition is employed to generate correlated random numbers based on the correlation matrix of the five underlying assets. Let $A \in \mathbb{R}^{5\times5}$ denote the correlation matrix among the five underlying assets, with entries ρ_{ij} representing the correlation between asset i and j. We assume that all the eigenvalues of a matrix A are positive. Then the matrix A is symmetric and positive definite [18]:

$$A = \begin{pmatrix} 1 & \rho_{12} & \rho_{13} & \rho_{14} & \rho_{15} \\ \rho_{12} & 1 & \rho_{23} & \rho_{24} & \rho_{25} \\ \rho_{13} & \rho_{23} & 1 & \rho_{34} & \rho_{35} \\ \rho_{14} & \rho_{24} & \rho_{34} & 1 & \rho_{45} \\ \rho_{15} & \rho_{25} & \rho_{35} & \rho_{45} & 1 \end{pmatrix}. \tag{2.1}$$

Cholesky decomposition is a matrix factorization technique applicable to symmetric positive definite matrices [19–21]. The matrix A is represented as the product of a lower triangular matrix L (Cholesky factor) and its transpose $L^\top$, i.e.,

$$A = LL^\top, \tag{2.2}$$

where L is detailed as follows:

$$L = \begin{pmatrix} L_{1,1} & 0 & 0 & 0 & 0 \\ L_{2,1} & L_{2,2} & 0 & 0 & 0 \\ L_{3,1} & L_{3,2} & L_{3,3} & 0 & 0 \\ L_{4,1} & L_{4,2} & L_{4,3} & L_{4,4} & 0 \\ L_{5,1} & L_{5,2} & L_{5,3} & L_{5,4} & L_{5,5} \end{pmatrix}. \tag{2.3}$$

Here, $L_{i,j}$ denotes the (i, j)-th entry of the lower triangular matrix L, with $(1 \leq i, j \leq 5)$. In this study, the matrix A represents the correlation matrix of five underlying assets and is symmetric and positive definite by construction. Therefore, we can apply Cholesky decomposition to A to obtain the lower triangular matrix L, which is used to generate correlated standard normal variables in the Monte Carlo simulation. The entries of L can be computed recursively: the diagonal elements use the first formula, and the off-diagonal elements use the second as

$$L_{i,i} = \sqrt{A_{i,i} - \sum_{k=1}^{i-1} L_{i,k}^2}, \quad L_{j,i} = \frac{1}{L_{i,i}} \left(A_{j,i} - \sum_{k=1}^{i-1} L_{j,k} L_{i,k} \right), \quad \text{for } j > i. \tag{2.4}$$

The entries of L are given in detail as follows:

$$L_{1,1} = 1, \quad L_{2,1} = \rho_{12}, \quad L_{2,2} = \sqrt{1 - \rho_{12}^2}, \quad L_{3,1} = \rho_{13}, \quad L_{3,2} = \frac{(\rho_{23} - \rho_{12}\rho_{13})}{\sqrt{1 - \rho_{12}^2}}, \quad L_{3,3} = \alpha, \tag{2.5}$$

$$L_{4,1} = \rho_{14}, \quad L_{4,2} = \frac{\rho_{24} - \rho_{12}\rho_{14}}{\sqrt{1 - \rho_{12}^2}}, \quad L_{4,3} = \frac{\beta}{\alpha}, \quad L_{4,4} = \sqrt{\delta - \left(\frac{\beta}{\alpha}\right)^2}, \quad L_{5,1} = \rho_{15}, \tag{2.6}$$

$$L_{5,2} = \frac{\rho_{25} - \rho_{12}\rho_{15}}{\sqrt{1 - \rho_{12}^2}}, \quad L_{5,3} = \frac{\gamma}{\alpha}, \quad L_{5,4} = \frac{\lambda - \frac{\beta\gamma}{\alpha^2}}{\sqrt{\delta - \left(\frac{\beta}{\alpha}\right)^2}}, \tag{2.7}$$

$$L_{5,5} = \sqrt{1 - \rho_{15}^2 - \frac{(\rho_{25} - \rho_{12}\rho_{15})^2}{1 - \rho_{12}^2} - \left(\frac{\gamma}{\alpha}\right)^2 - \frac{\left(\lambda - \frac{\beta\gamma}{\alpha^2}\right)^2}{\delta - \left(\frac{\beta}{\alpha}\right)^2}}, \tag{2.8}$$

where the intermediate variables $\alpha, \beta, \delta, \gamma, \lambda$ are described as

$$\alpha = \sqrt{1 - \rho_{13}^2 - \frac{(\rho_{23} - \rho_{12}\rho_{13})^2}{1 - \rho_{12}^2}}, \quad \beta = \rho_{34} - \rho_{13}\rho_{14} - \frac{(\rho_{23} - \rho_{12}\rho_{13})(\rho_{24} - \rho_{12}\rho_{14})}{1 - \rho_{12}^2}, \tag{2.9}$$

$$\delta = 1 - \rho_{14}^2 - \frac{(\rho_{24} - \rho_{12}\rho_{14})^2}{1 - \rho_{12}^2}, \quad \gamma = \rho_{35} - \rho_{13}\rho_{15} - \frac{(\rho_{23} - \rho_{12}\rho_{13})(\rho_{25} - \rho_{12}\rho_{15})}{1 - \rho_{12}^2}, \tag{2.10}$$

$$\lambda = \rho_{45} - \rho_{14}\rho_{15} - \frac{(\rho_{24} - \rho_{12}\rho_{14})(\rho_{25} - \rho_{12}\rho_{15})}{1 - \rho_{12}^2}. \tag{2.11}$$

The lower triangular matrix L is used to convert a vector of independent standard normal variables

into a vector of correlated standard normal variables:

$$\begin{pmatrix} Z_1^* \\ Z_2^* \\ Z_3^* \\ Z_4^* \\ Z_5^* \end{pmatrix} = L \begin{pmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \\ Z_5 \end{pmatrix} = \begin{pmatrix} L_{1,1} & 0 & 0 & 0 & 0 \\ L_{2,1} & L_{2,2} & 0 & 0 & 0 \\ L_{3,1} & L_{3,2} & L_{3,3} & 0 & 0 \\ L_{4,1} & L_{4,2} & L_{4,3} & L_{4,4} & 0 \\ L_{5,1} & L_{5,2} & L_{5,3} & L_{5,4} & L_{5,5} \end{pmatrix} \begin{pmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \\ Z_5 \end{pmatrix}. \tag{2.12}$$

The explicit form is given by:

$$Z_1^* = Z_1, \;\; Z_2^* = \rho_{12} Z_1 + \sqrt{1-\rho_{12}^2} Z_2, \;\; Z_3^* = \rho_{13} Z_1 + \frac{(\rho_{23} - \rho_{12}\rho_{13})}{\sqrt{1-\rho_{12}^2}} Z_2 + \alpha Z_3, \tag{2.13}$$

$$Z_4^* = \rho_{14} Z_1 + \frac{\rho_{24} - \rho_{12}\rho_{14}}{\sqrt{1-\rho_{12}^2}} Z_2 + \frac{\beta}{\alpha} Z_3 + \sqrt{\delta - \left(\frac{\beta}{\alpha}\right)^2} Z_4, \tag{2.14}$$

$$Z_5^* = \rho_{15} Z_1 + \frac{\rho_{25} - \rho_{12}\rho_{15}}{\sqrt{1-\rho_{12}^2}} Z_2 + \frac{\gamma}{\alpha} Z_3 + \frac{\lambda - \frac{\beta\gamma}{\alpha^2}}{\sqrt{\delta - \left(\frac{\beta}{\alpha}\right)^2}} Z_4 + \sqrt{1 - \rho_{15}^2 - \frac{(\rho_{25} - \rho_{12}\rho_{15})^2}{1-\rho_{12}^2} - \left(\frac{\gamma}{\alpha}\right)^2 - \frac{\left(\lambda - \frac{\beta\gamma}{\alpha^2}\right)^2}{\delta - \left(\frac{\beta}{\alpha}\right)^2}} Z_5, \tag{2.15}$$

where Z_1, Z_2, Z_3, Z_4, and Z_5 are identically independent standard normal distributions.

Using these transformed values, we shall simulate the asset prices under geometric Brownian motion. To price the fair value of the five-asset step-down ELS applying the Monte Carlo simulation, let us consider the following discrete stock paths:

$$S_j(t_{i+1}) = S_j(t_i) e^{\left(r - 0.5\sigma_j^2\right)\Delta t + \sigma_j \sqrt{\Delta t} Z_j^*(t_i)}, \quad j = 1, 2, 3, 4, 5, \tag{2.16}$$

where $t_i = i\Delta t$, r is the risk-free interest, $\Delta t = 1/360$ is the time step size, and σ_j denotes the volatility of the j-th underlying asset, with $j = 1, 2, 3, 4, 5$. This computation is repeated over multiple time steps and across a large number of paths in the Monte Carlo framework. The resulting paths are then used to compute the fair price and risk profile of the structured ELS product. For a more comprehensive discussion of the implementation details, including algorithmic procedures and parameter configurations, the reader is referred to [14,22], which provide in-depth treatments of similar Monte Carlo simulation frameworks applied to four-asset Equity-Linked Securities.

3. Computational tests

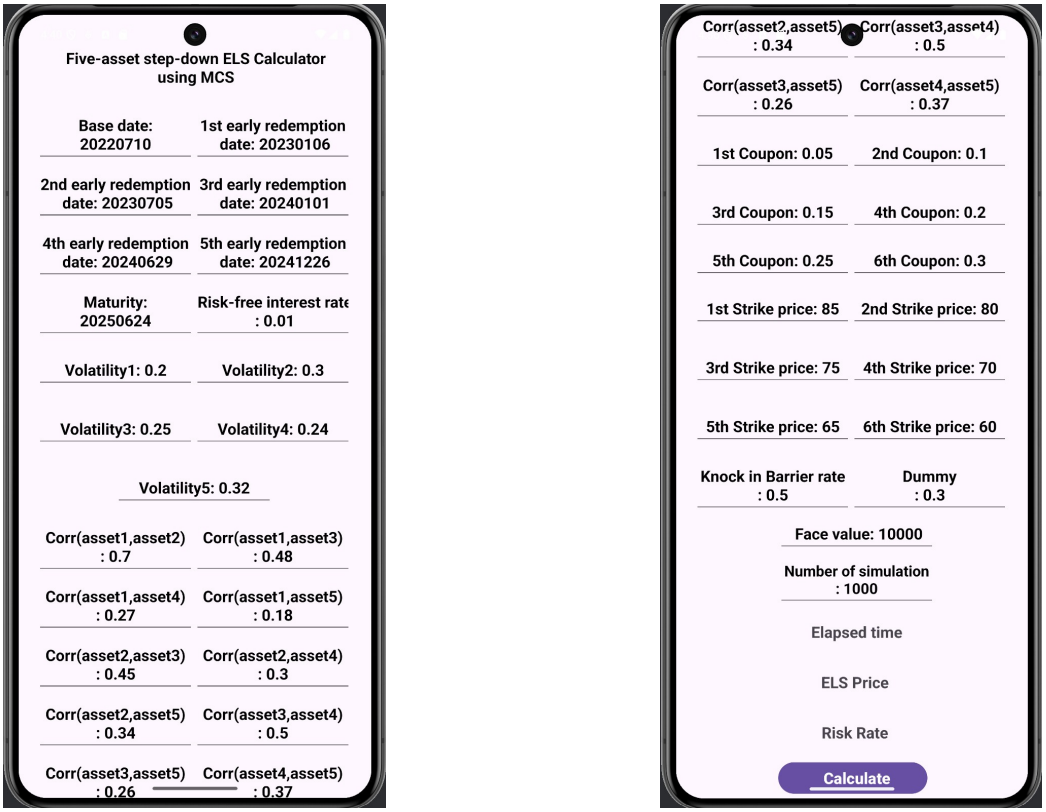
We shall now perform computational tests to implement the proposed pricing algorithm on the Android platform. The computational results will be obtained using the proposed Android calculator on a Pixel 8 Pro device with API level 33 of the Android platform, equipped with a 1.82 GHz hexa-core processor and 1.5 GB of RAM. The graphical results will be generated using MATLAB version R2024a on a PC with an Intel(R) Core(TM) i7-12700 CPU @ 2.10 GHz and 16 GB of RAM.

Unless explicitly specified, the parameters listed in Table 1 are employed throughout this section.

Table 1. A list of parameters used in Section 3.

Parameters	Values
Repayment dates, $[T_1, T_2, T_3, T_4, T_5, T_6]$	$[0.5, 1, 1.5, 2, 2.5, 3]$
Maturity date, $T = T_6$	3
Time step size, Δt	$1/360$
The number of total time step, $N_t = T/\Delta t$	1080
Volatilities of five underlying assets, $[\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6]$	$[0.2, 0.3, 0.25, 0.24, 0.32]$
Strike price ratios, $[K_1, K_2, K_3, K_4, K_5, K_6]$	$[0.85, 0.8, 0.75, 0.7, 0.65, 0.6]$
Knock-in barrier, kib	0.5
Coupon rates on redemption dates, $[c_1, c_2, c_3, c_4, c_5, c_6]$	$[0.05, 0.1, 0.15, 0.2, 0.25, 0.3]$
Dummy rate on maturity date, d	0.3
Initial prices of five underlying assets, $[X_1(0), X_2(0), X_3(0), X_4(0), \text{and } X_5(0)]$	$[100, 100, 100, 100, 100]$
Face value, F	$10,000$
Interest rate, r	0.01
The number of simulations, N_p	10^6
Correlation coefficients matrix, A	$\begin{pmatrix} 1 & 0.7 & 0.48 & 0.27 & 0.18 \\ 0.7 & 1 & 0.45 & 0.3 & 0.34 \\ 0.48 & 0.45 & 1 & 0.5 & 0.26 \\ 0.27 & 0.3 & 0.5 & 1 & 0.37 \\ 0.18 & 0.34 & 0.26 & 0.37 & 1 \end{pmatrix}$

Figure 1 displays the main interface of the Android-based ELS pricing application developed in this study. Figure 1(a) and Figure 1(b) show the user input screen, where one can specify key simulation parameters such as early redemption dates, maturity, interest rate, volatilities, correlation coefficients, coupon rates, strike prices, knock-in barrier rate, dummy rate, face value, and number of simulation. The bottom part of Figure 1(b) shows the resulting fair price, risk rate, and elapsed computation time, which are displayed after the user presses the "Calculate" button. The computational results vary based on the input data, which varies with the simulation count.



(a) (b)

Figure 1. Calculator for pricing five-asset ELS on the Android platform.

Now, we provide the pseudo algorithm. In the initialization phase, the user inputs the initial prices of the five underlying assets, the strike conditions including early redemption dates and knock-in barriers, and the parameters required to simulate their stochastic dynamics, such as volatilities and correlation coefficients. Algorithm 1 describes the step-by-step procedure to price a five-asset step-down ELS using a Monte Carlo simulation framework:

1. **Step 1: Cholesky decomposition of the correlation matrix.**
   First, we construct the correlation matrix A and apply the Cholesky factorization, yielding a lower triangular matrix L. This matrix L is then used to generate correlated random draws from independent standard normal variables, ensuring that the correlation structure of underlying assets is accurately reflected in the simulation.
2. **Step 2: Monte Carlo path generation for underlying assets.**
   The main loop iterates over N_p Monte Carlo trials. In each trial, we generate daily price paths for all five underlying assets by discretizing a geometric Brownian motion. Specifically:
   - Compute $Z^* = LZ$, where Z is a vector of five i.i.d. $\mathcal{N}(0, 1)$ random variables.
   - Update the asset prices at each discrete time step Δt according to

$$S_j(t_{i+1}) = S_j(t_i)e^{\left(r-0.5\sigma_j^2\right)\Delta t+\sigma_j\sqrt{\Delta t}Z_j^*(t_i)}, \quad j = 1, 2, 3, 4, 5. \tag{3.1}$$

   These paths incorporate both drift and volatility, as well as the correlation captured by the Cholesky factorization.

3. **Step 3: Worst performer and early redemption.**
   After generating a full set of asset paths for a single trial, the algorithm identifies the worst performer (WP) by taking the minimum of all five asset ratios (i.e., S_1, S_2, S_3, S_4, S_5) at each coupon date. It then checks whether WP exceeds each corresponding strike level K_i on early redemption dates T_i:

   - If $\text{WP}(T_i) \geq K_i$, the ELS is assumed to redeem early, and the coupon c_i (plus principal) is credited to the payoff M_i.
   - If no early redemption occurs through all prescribed dates, the algorithm checks whether the knock-in barrier has been breached at any time. If a breach has occurred, the final payoff is determined by the terminal value of the worst performer. Otherwise, if the barrier has not been touched, the payoff is adjusted by a "dummy rate" coupon rate (e.g., $1 + d$).

4. **Step 4: Discounting and averaging over simulations.**
   Each realized payoff M_i (or final payoff M_6) is then *discounted* to present value using the risk-free rate r. Finally, the average across all N_p trials is computed to obtain the *estimated fair price* of the five-asset step-down ELS: $V_0 = \dfrac{1}{N_p} \displaystyle\sum_{k=1}^{6} e^{-r\tau_k} \textbf{Payoff}_k$ where τ_k is the relevant redemption (or maturity) time for the k-th trial. Simultaneously, the *Risk rate*, corresponding to the probability of knock-in, is approximated by counting how many paths out of N_p breach the barrier at least once: $\textit{Risk rate} = \frac{\text{Number of knock-in events}}{N_p}$. This statistic quantifies the likelihood of incurring principal loss due to the knock-in condition.

In summary, Algorithm 1 offers a systematic approach for pricing multi-asset step-down ELS with knock-in barriers. By incorporating correlated geometric Brownian motions (via Cholesky decomposition) and explicitly modeling the boundary conditions (early-redemption and knock-in features), it captures the key structural characteristics of these structured products. Monte Carlo simulation, while computationally intensive, is particularly suitable for products whose payoffs depend on the paths of multiple underlying assets. Moreover, this algorithm can be readily implemented on mobile devices, making advanced derivative pricing more accessible for practitioners and retail investors. The knock-in probability (*Risk rate*) further aids in quantifying downside exposure, shedding light on the product's risk-reward profile and assisting both issuers and investors in assessing its attractiveness.

Algorithm 1 Pricing algorithm for five-assets step-down ELS using MCS

Require:

Redemption dates (dates when redemption is possible) : $T_1, T_2, T_3, T_4, T_5, T_6$

Maturity date (final contract expiration date) : $T = T_6$

Time step size (discrete time increment for simulation) : $\Delta t = 1/360$

The number of total time step (total number of time steps) : $N_t = T/\Delta t$

Volatilities of five underlying assets : $\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5$

Strike price ratios (threshold levels for redemption evaluation) : $K_1, K_2, K_3, K_4, K_5, K_6$

Knock-in barrier (barrier level for knock-in condition) : kib

Coupon rates on redemption dates (fixed returns paid at redemption dates) : $c_1, c_2, c_3, c_4, c_5, c_6$

Dummy rate on maturity date (notional rate applied only at maturity to adjust the final payoff) : d

Initial prices of five underlying assets : $X_1(0)$, $X_2(0)$, $X_3(0)$, $X_4(0)$, and $X_5(0)$

Scale the underlying assets with the initial price

: $S_1(t_1) = X_1(t_1)/X_1(0)$, $S_2(t_1) = X_2(t_1)/X_2(0)$, $S_3(t_1) = X_3(t_1)/X_3(0)$, $S_4(t_1) = X_4(t_1)/X_4(0)$, and $S_5(t_1) = X_5(t_1)/X_5(0)$, where $t_1 = 0$

Face value (principal amount of the contract) : F

Interest rate (risk-free continuous interest rate) : r

The number of simulations (number of Monte Carlo sample paths) : N_p

Correlation coefficients matrix : $A = \begin{pmatrix} 1 & \rho_{12} & \rho_{13} & \rho_{14} & \rho_{15} \\ \rho_{12} & 1 & \rho_{23} & \rho_{24} & \rho_{25} \\ \rho_{13} & \rho_{23} & 1 & \rho_{34} & \rho_{35} \\ \rho_{14} & \rho_{24} & \rho_{34} & 1 & \rho_{45} \\ \rho_{15} & \rho_{25} & \rho_{35} & \rho_{45} & 1 \end{pmatrix}$ (Eq. (2.1))

▹ **Step 1.** Cholesky decomposition of a (5×5) correlation matrix A is given by $A = LL^T$, where L is a lower triangular matrix given by $L = \begin{pmatrix} L_{1,1} & 0 & 0 & 0 & 0 \\ L_{2,1} & L_{2,2} & 0 & 0 & 0 \\ L_{3,1} & L_{3,2} & L_{3,3} & 0 & 0 \\ L_{4,1} & L_{4,2} & L_{4,3} & L_{4,4} & 0 \\ L_{5,1} & L_{5,2} & L_{5,3} & L_{5,4} & L_{5,5} \end{pmatrix}$ (Eq. (2.3))

Set payment on redemption dates $M_1 = M_2 = M_3 = M_4 = M_5 = M_6 = 0$

Set $risk$ (the number of times the knock-in barrier is breached) $risk = 0$

for $iteration = 1$ to N_p **do**

▹ **Step 2.** Generate daily discrete stock paths for t_j using the Monte Carlo method

for $j = 1$ to N_t **do**

$\left(Z_1^*, Z_2^*, Z_3^*, Z_4^*, Z_5^*\right)^\top = L \times \left(Z_1, Z_2, Z_3, Z_4, Z_5\right)^\top, \quad Z_n \sim N(0,1), \quad (n = 1, 2, 3, 4, 5)$ (Eq. (2.12))

$S_n(t_{j+1}) = S_n(t_j) \exp\left(\left(r - 0.5\sigma_n^2\right)\Delta t + \sigma_n \sqrt{\Delta t} Z_n{}^*(t_j)\right)$ (Eq. (3.1))

end for

▹ Define worst performer value

$WP = \min([S_1, S_2, S_3, S_4, S_5])$

▹ **Step 3.** Check the values of discrete stock paths at checking days

if $WP(T_1) \geq K_1$ **then** $M_1 = M_1 + (1 + c_1)F$

else if $WP(T_2) \geq K_2$ **then** $M_2 = M_2 + (1 + c_2)F$

else if $WP(T_3) \geq K_3$ **then** $M_3 = M_3 + (1 + c_3)F$

else if $WP(T_4) \geq K_4$ **then** $M_4 = M_4 + (1 + c_4)F$

else if $WP(T_5) \geq K_5$ **then** $M_5 = M_5 + (1 + c_5)F$

else if $WP(T_6) \geq K_6$ **then** $M_6 = M_6 + (1 + c_6)F$

else if $\min\limits_{1 \leq j \leq N_t} \{WP(T_j)\} \leq kib$ **then** $M_6 = M_6 + WP(t_{N_t})F$

$risk = risk + 1$

else

$M_6 = M_6 + (1 + d)F$

end if

end for

▹ **Step 4.** Discount to present price and take average

$Price = \sum_{i=1}^{6} \left(e^{-rT_i} M_i / N_p\right)$

$Risk\ rate = risk/N_p$

Figure 2 illustrates both the convergence behavior of our Monte Carlo simulations for four-asset and five-asset ELS pricing and the associated computational costs. Specifically, Figure 2(a) shows the fair-price estimates as a function of the total number of simulated paths N_p, plotted for each of the two product structures (four-asset vs. five-asset). In Figure 2(a), for each number of paths, we display results from 20 simulations. The vertical spread indicates the variance or confidence interval in the estimates across multiple independent runs. The plot demonstrates that, as N_p increases, the simulated ELS prices converge toward stable values, confirming the robustness and consistency of the underlying Monte Carlo approach. In Figure 2(b), the measured *elapsed times* (in seconds) for performing the full pricing procedure are depicted, again for both four- and five-asset configurations. This highlights the computational overhead introduced by additional underlying assets. As expected, the five-asset ELS requires more simulation steps for generating and processing correlated asset paths, resulting in a longer runtime. Finally, Figure 2(c) shows the *ratio* of the elapsed times between the five-asset and four-asset cases, providing a more direct comparison of the relative computational expense. The result demonstrates that the computational cost increases nonlinearly with the number of underlying assets, suggesting that the correlation structure and the higher-dimensional random draws impose additional complexity. Overall, the computational results in Figure 2 confirm both the efficiency and accuracy of our Monte Carlo framework, while also underscoring the trade-off between realism (more underlying assets) and computational time in practical ELS pricing scenarios.

Figure 3 depicts the evolution of the worst performer (WP) over the number of checking days, for cases ranging from one to five underlying assets. The WP at each time step is defined as the minimum scaled value among all included underlying assets. Each colored curve corresponds to a different total number of underlying assets, highlighting that as the asset count grows, the WP trajectory generally shifts downward. This behavior stems from the increased probability that at least one asset will underperform when more underlying assets are included. Thus, a five-asset configuration typically exhibits a lower WP than scenarios with fewer underlying assets. This phenomenon implies a higher likelihood of crossing the knock-in barrier, reflecting the inherent trade-off between offering higher coupons and exposing investors to greater downside risk.

We calculated and compared risk rates for each number of underlying assets and each number of trials in Figure 4. The risk rate is the number of times the knock-in-barrier is hit divided by the number of trials. Each point is the average of values calculated 10 times in MATLAB.

As shown in Figure 4, the variance of the estimated risk rate decreases as the number of simulation paths N_p increases, and the estimate gradually converges. We further confirm that the risk rate generally increases with the number of underlying assets, since the worst performer condition is more readily satisfied, making it easier to breach the knock-in barrier *kib*. This aligns with practical financial intuition: adding more underlying assets can offer a higher coupon but also entails greater knock-in (and thus principal loss) risk. The risk rate is defined as the ratio of "the number of simulations in which the knock-in barrier is touched at least once" to "the total number of simulations". Mathematically, if kib_n is the number of knock-in occurrences out of N_p total Monte Carlo trials, then the estimated risk rate $\hat{p}$ is

$$\hat{p} = \frac{kib_n}{N_p}. \tag{3.2}$$

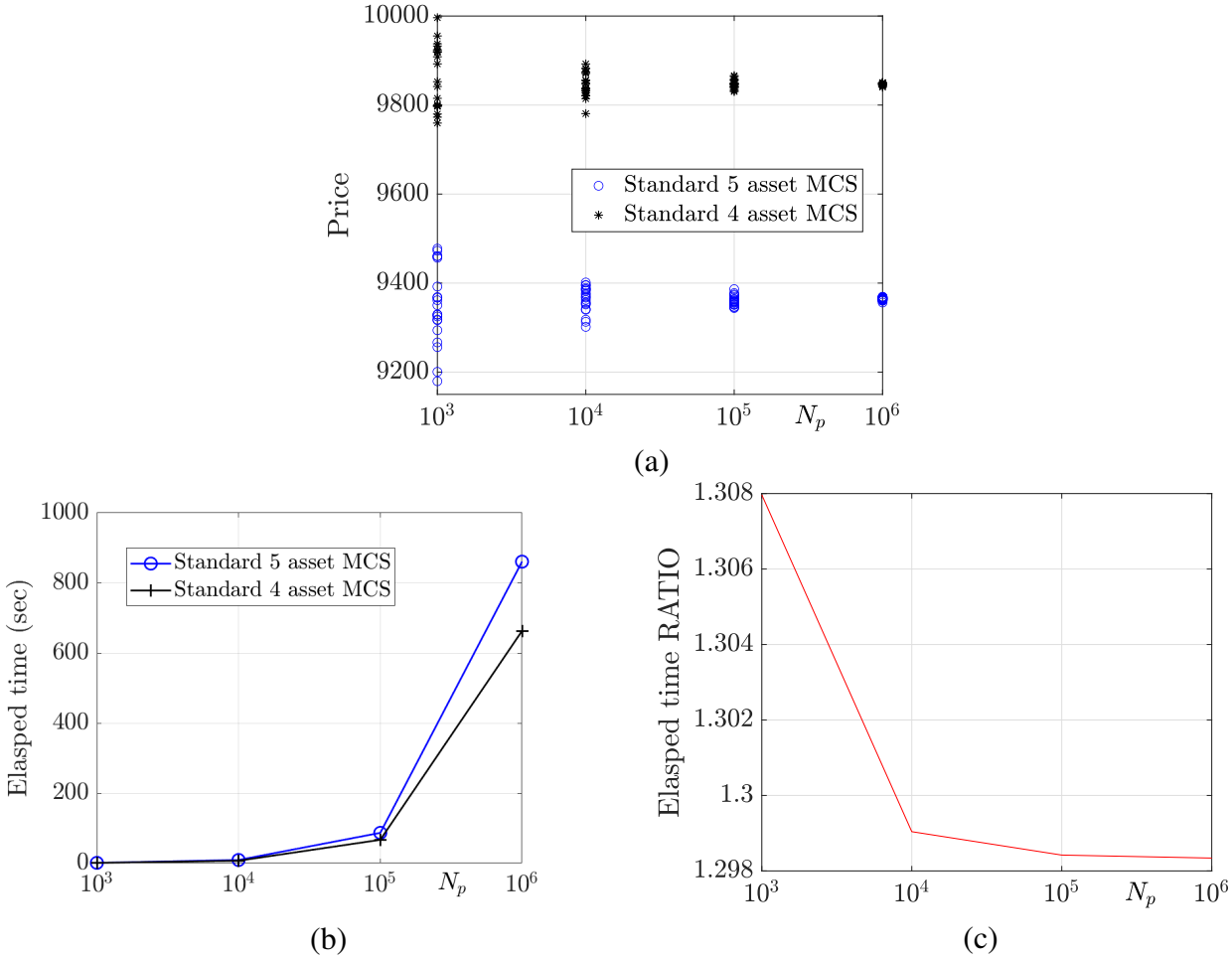


Figure 2. (a) Convergence of the Monte Carlo-estimated ELS prices for four-asset and five-asset products as the number of simulated paths N_p increases. (b) The elapsed times of four-asset and five-asset ELS pricing. (c) The elapsed time ratio of four-asset and five-asset ELS pricing.

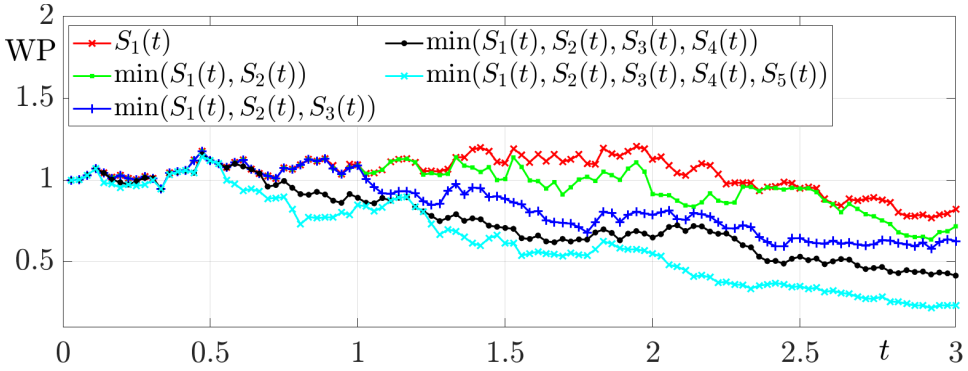


Figure 3. Worst performer (WP) trajectories for one through five underlying assets, plotted over the number of checking days. Each colored line corresponds to the WP for a different number of underlying assets.

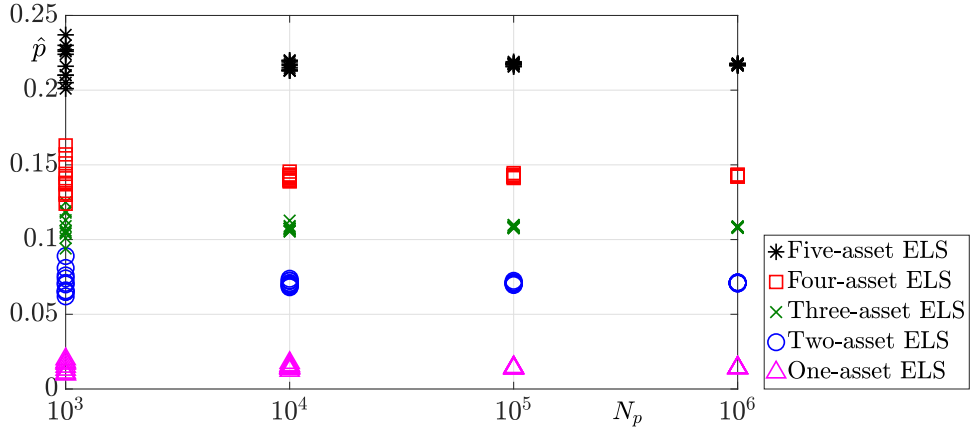


Figure 4. Estimated knock-in risk rates (for short, Risk rate) across varying numbers of Monte Carlo simulation paths and different counts of underlying assets.

Given that kib_n follows a binomial (N_p, p) distribution, $\hat{p}$ is the sample mean of this binomial random variable. For sufficiently large N_p, the Central Limit Theorem implies that $\hat{p}$ can be approximated by a normal distribution with mean p and standard error $\mathcal{SE}$:

$$\mathcal{SE}(\hat{p}) \approx \sqrt{\frac{\hat{p}\,(1-\hat{p})}{N_p}}. \tag{3.3}$$

For instance, if $N_p = 10^6$ and the estimated risk rate is $\hat{p} = 0.20$, then the standard error is approximately

$$\mathcal{SE}(\hat{p}) \approx \sqrt{\frac{0.22 \times 0.78}{1,000,000}} = 0.0004, \tag{3.4}$$

yielding a 95% confidence interval of about $0.20 \pm 2 \times 0.0004$. This is indicates that with a sufficiently large number of simulations, the risk rate can be estimated accurately, and the confidence interval for $\hat{p}$ becomes very narrow. Meanwhile, from an investor's perspective, the risk rate indicates the likelihood of breaching the knock-in barrier, thereby signaling the potential for principal loss. For example, if a five-asset ELS exhibits an estimated risk rate of 0.20, it suggests a 20% chance that the product will transition into a knock-in state, exposing the investor to greater downside. Consequently, investors must weigh "the higher coupon benefit" against "the increased risk of principal loss". In addition, statistical analysis of the risk rate is of considerable value for product structuring. A securities issuer aiming to maintain the risk rate below a certain threshold, for instance, can adjust the composition of underlying assets S_i, the knock-in barrier kib, the strike K_i, or the coupon rates c_i. Conversely, if the issuer focuses on maximizing coupon rates (accepting higher risk), such parameter adjustments can also be explored accordingly.

Table 2 shows the computed fair prices of the step-down ELS for varying numbers of underlying assets, ranging from one to five. The ELS prices are calculated in MATLAB. Each ELS price represents the average of 10 simulation runs, each based on $N_p = 10^6$ Monte Carlo simulations. Each entry corresponds to the Monte Carlo-estimated price under identical model settings (e.g., the same knock-in barrier, coupon rates, and correlation assumptions, etc.), with only the number of underlying assets changing. The results indicate a notable decline in the ELS price as the number of underlying assets

increases. This pattern can be attributed primarily to the worst performer effect, whereby a greater number of assets heightens the likelihood that at least one asset will underperform, thereby reducing the product's final payoff. Such findings underscore the trade-off between offering a potentially higher coupon by including multiple underlying assets and incurring a greater risk of principal loss from knock-in events.

Table 2. Computed ELS fair prices as the number of underlying assets increases from one to five.

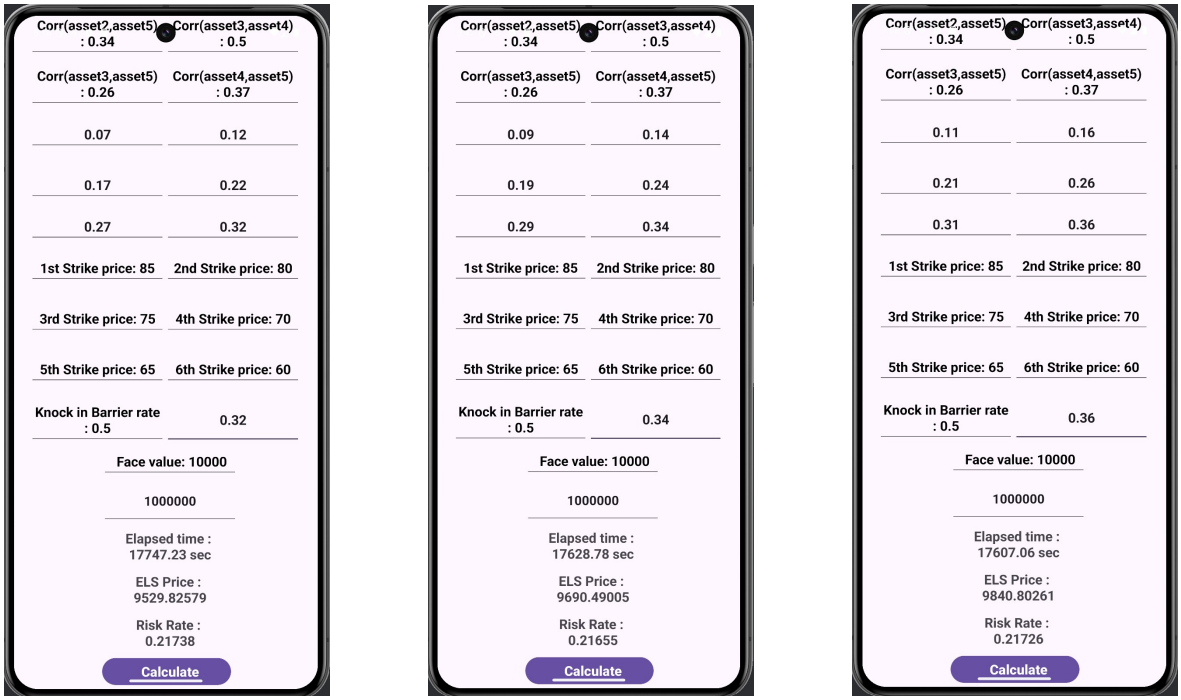
Number of assets	1	2	3	4	5
ELS price	10465.979	10188.187	10012.546	9842.888	9375.536

Table 3 reports the fair ELS prices under three different increments of the coupon rate (increases of 0.02, 0.04, and 0.06, respectively). For each scenario, the same baseline parameters remain fixed, ensuring that the only varying factor is the coupon rate. The prices were computed by averaging the results of multiple Monte Carlo simulations (e.g., five independent runs of 10^6 paths each) to improve the reliability of the estimates. A notable trend is that, as the coupon rate is increased, the fair price of the ELS decreases. This reflects the heightened risk borne by the issuer, as a higher coupon raises potential payouts to investors but also reduces the product's market-consistent value. From an investor's perspective, a higher coupon is attractive yet comes with a potentially higher probability of knock-in if the underlying assets perform poorly. Meanwhile, from an issuer's perspective, the lower fair price indicates the increased cost associated with providing a richer payoff structure. This trade-off underscores the importance of selecting an optimal coupon rate that balances investment appeal with risk and pricing considerations.

Table 3. Fair prices for step-down five-asset ELS with the knock-in barrier condition, under three different incremental increases in the coupon rate (0.02, 0.04, and 0.06).

Increase in each coupon rate and dummy rate	0.02	0.04	0.06
ELS price	9528.42	9681.53	9839.27

Figure 5 presents screenshots of the proposed Android-based ELS pricing calculator, focusing on how the computational results of fair price as the coupon rates and dummy rate values are incrementally increased. In Figure 5(a), Figure 5(b), and Figure 5(c), each coupon rate and dummy rate value are raised by 0.02, 0.04, and 0.06, respectively, compared to their baseline values. As these values increase, the application displays higher fair prices and updates the corresponding risk rates in real time. This behavior reflects the structural properties of multi-asset step-down ELS, where increasing coupon rates and dummy rate components generally enhances the potential payoff to investors, thus leading to a higher estimated product value. The figure also highlights that the calculator interface conveniently reports the total computation time, demonstrating that even with added complexity such as elevated coupon rates or larger dummy rate values, real-time simulation on a mobile device is both feasible and user-friendly.



(a) (b) (c)

Figure 5. Calculator for pricing five-asset ELS with increased coupon and dummy rates. Each coupon rate and dummy rate are increased by 0.02, 0.04, and 0.06 in (a), (b), and (c), respectively.

4. Discussion

Despite the strengths of the proposed framework, several limitations should be acknowledged. First, as five-asset ELS products are not yet widely available in the financial market, and empirical validation using real transaction data was not feasible. Therefore, the computational experiments in this study were conducted using synthetic but structurally realistic input parameters, inspired by commonly traded ELS products with fewer underlying assets. Although this limits the ability to directly assess real-world pricing accuracy, the proposed framework is designed to be adaptable and applicable to actual market data when such products become available in the future.

Future research could extend the current model by incorporating fractional Brownian motion, regime-switching dynamics, time-varying volatilities, or jump-diffusion processes to better reflect market complexities. Additionally, integrating machine learning techniques or variance reduction methods may further improve the efficiency and generalizability of the pricing algorithm. Evaluating the robustness of the framework across various market conditions will also be an important step toward real-world deployment of the proposed mobile-based pricing tool.

5. Conclusions

In this study, we developed a robust and efficient Monte Carlo simulation framework for pricing five-asset ELS with step-down and knock-in barrier structures. The robustness of the proposed method

is supported by convergence analyses of simulated ELS prices and risk rates across varying numbers of assets and simulation paths, as well as by the precise modeling of asset correlations via Cholesky decomposition. The efficiency of the framework is demonstrated through its successful implementation on an Android platform, enabling real-time computations on resource-constrained mobile devices. This highlights the feasibility of performing complex financial calculations in portable environments, thereby enhancing the accessibility of advanced pricing tools. A key contribution of this work is the practical deployment of the pricing algorithm on a mobile platform, allowing for real-time evaluation of both fair price and associated risk. Through extensive computational experiments, we examined the convergence behavior of the simulation, evaluated the sensitivity of ELS prices to changes in the number of underlying assets and coupon rates, and analyzed the corresponding risk profiles. The results consistently show that an increase in the number of underlying assets leads to lower ELS prices and elevated risk exposure, primarily due to the worst performer effect. The Android-based pricing tool enhances accessibility and usability for both individual investors and financial institutions, supporting more informed investment decisions. Overall, this study not only advances numerical techniques in multi-asset derivative pricing but also shows how mobile-based Fintech solutions can deliver sophisticated quantitative methods that enhance transparency, accessibility, and investor education in evaluating structured financial products, thereby strengthening confidence in complex instruments and advancing the Fintech agenda.

Author contributions

Junseok Kim: Funding acquisition, Investigation, Methodology, Project administration, Validation, Writing–review and editing; Juho Ma: Data curation, Formal analysis, Investigation, Methodology, Resources, Software, Validation, Visualization, Writing–original draft, Writing–review and editing; Hyunho Shin: Formal analysis, Investigation, Software, Validation, Visualization, Writing–original draft, Writing–review and editing; Hyundong Kim: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Software, Supervision, Validation, Writing–original draft, Writing–review and editing;

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The authors gratefully acknowledge the insightful comments and constructive suggestions from the reviewers and the editor. This study was supported by the 2023 Academic Research Support Program in Gangneung-Wonju National University.

Conflict of interest

Hyundong Kim is the Guest editor for AIMS Mathematics and was not involved in the editorial review or the decision to publish this article. All authors declare that there are no competing interests.

The authors have no conflicts of interest to declare.

References

1. H. Kim, S. Park, K. S. Moon, Markov regime-switching in pricing equity-linked securities: An empirical study for losses in HSCEI-linked products, *Financ. Res. Lett.*, **76** (2025) 106929. https://doi.org/10.1016/j.frl.2025.106929
2. R. K. Maurya, D. Li, A. P. Singh, V. K. Singh, Numerical algorithm for a general fractional diffusion equation, *Math. Comput. Simul.*, **223** (2024), 405–432. https://doi.org/10.1016/j.matcom.2024.04.018
3. A. P. Singh, P. Rajput, R. K. Maurya, V. K. Singh, High order stable numerical algorithms for generalized time-fractional deterministic and stochastic telegraph models, *Comput. Appl. Math.*, **43** (2024), 401. https://doi.org/10.1007/s40314-024-02900-6
4. A. P. Singh, R. K. Maurya, V. K. Singh, Analysis of a robust implicit scheme for space-time fractional stochastic nonlinear diffusion wave model, *Int. J. Comput. Math.*, **100** (2023), 1625–1645. https://doi.org/10.1080/00207160.2023.2207677
5. P. Roul, V. P. Goura, A compact finite difference scheme for fractional Black–Scholes option pricing model, *Appl. Numer. Math.*, **166** (2021), 40–60. https://doi.org/10.1016/j.apnum.2021.03.017
6. J. Wang, Y. Yan, W. Chen, W. Shao, W. Tang, Equity-linked securities option pricing by fractional Brownian motion, *Chaos Soliton. Fract.*, **144** (2021), 110716. https://doi.org/10.1016/j.chaos.2021.110716
7. S. T. Kim, H. G. Kim, J. H. Kim, ELS pricing and hedging in a fractional Brownian motion environment, *Chaos Soliton. Fract.*, **142** (2021), 110453. https://doi.org/10.1016/j.chaos.2020.110453
8. J. Wang, J. Ban, J. Han, S. Lee, D. Jeong, Mobile platform for pricing of Equity-Linked Securities, *J. Korean Soc. Ind. Appl. Math.*, **21** (2017), 181–202. http://dx.doi.org/10.12941/jksiam.2017.21.181
9. J. Kim, D. Jeong, H. Jang, H. Kim, Mobile APP for computing option price of the four-underlying asset step-down ELS, *J. Korean Soc. Ind. Appl. Math.*, **4** (2022), 343–352. http://doi.org/10.12941/jksiam.2022.26.343
10. H. Jang, H. Han, H. Park, W. Lee, J. Lyu, J. Park, et al., Android application for pricing two- and three-asset Equity-Linked Securities, *J. Korean Soc. Ind. Appl. Math.*, **23** (2019), 237–251. http://doi.org/10.12941/jksiam.2019.23.237
11. S. Kim, J. Lyu, W. Lee, E. Park, H. Jang, C. Lee, et al., A Practical Monte Carlo Method for Pricing Equity-Linked Securities with Time-Dependent Volatility and Interest Rate, *Comput. Econ.*, **63** (2024), 2069–2086. https://doi.org/10.1007/s10614-023-10394-3
12. V. Podlozhnyuk, M. Harris, Monte carlo option pricing, *CUDA SDK*, (2008).
13. N. Stamatopoulos, D. J. Egger, Y. Sun, C. Zoufal, R. Iten, N. Shen, et al., Option pricing using quantum computers, *Quantum*, **4** (2020), 291. https://doi.org/10.22331/q-2020-07-06-291

14. H. Hwang, Y. Choi, S. Kwak, Y. Hwang, S. Kim, J. Kim, Efficient and accurate finite difference method for the four underlying asset ELS, *Pure Appl. Math.*, **28** (2021), 329–341. https://doi.org/10.7468/jksmeb.2021.28.4.329

15. H. Jang, S. Kim, J. Han, S. Lee, J. Ban, H. Han, et al., Fast Monte Carlo simulation for pricing equity-linked securities, *Comput. Econ.*, **56** (2020), 865–882. https://doi.org/10.1007/s10614-019-09947-2

16. C. Yoo, Y. Choi, S. Kim, S. Kwak, Y. Hwang, J. Kim, Fast Pricing of Four Asset Equity-Linked Securities Using Brownian Bridge, *J. Korean Soc. Ind. Appl. Math.*, **25** (2021), 82–92. https://doi.org/10.12941/jksiam.2021.25.082

17. H. Jang, H. Kim, S. Jo, H. Kim, S. Lee, J. Lee, et al., Fast ANDROID implementation of Monte Carlo simulation for pricing equity-linked securities, *J. Korean Soc. Ind. Appl. Math.*, **24** (2020), 79–84. http://dx.doi.org/10.12941/jksiam.2020.24.079

18. C. R. Johnson, Positive definite matrices, *Am. Math. Mon.*, **77** (1970), 82–92. https://doi.org/10.1080/00029890.1970.11992465

19. P. Glasserman, Monte Carlo methods in financial engineering, *New York: Springer.*, **53** (2004). https://doi.org/10.1007/978-0-387-21617-1

20. J. Wang, C. Liu, Generating multivariate mixture of normal distributions using a modified Cholesky decomposition, *Proc. 38th Conf. Winter Simul.*, (2006). https://doi.org/10.1109/WSC.2006.323100

21. T. Wang, Q. Huang, A new Newton method for convex optimization problems with singular Hessian matrices, *AIMS Math.*, **8** (2023). 21161–21175. https://doi.org/10.3934/math.20231078

22. J. Kim, D. Jeong, H. Jang, H. Kim, Mobile APP for computing option price of the four-underlying asset step-down ELS, *J. Korean Soc. Ind. Appl. Math.*, **26** (2022), 343–352. http://doi.org/10.12941/jksiam.2022.26.343

Appendix

The following MATLAB source code computes the risk rate and determines the fair price of a five-asset step-down ELS with a knock-in barrier condition:

```
clear all;
T=3; dt=1/360; N=round(T/dt);
coupon=[0.05 0.1 0.15 0.2 0.25 0.3];
step=[1 2 3 4 5 6]*N/6;
S_rate=[0.85 0.80 0.75 0.70 0.65 0.60];
dummy=0.3; kib=0.5; ns=1.0e+6; face=10000; S=[1 1 1 1 1]; r=0.01;
vol=[0.2 0.3 0.25 0.24 0.32];
rho=[0.7 0.48 0.27 0.18 0.45 0.3 0.34 0.5 0.26 0.37];
stepn=length(step); strike_ch(1,stepn)=0;
payoff(1,stepn)=0; dsum(1,stepn)=0;
disc_payoff(1,stepn)=0; payment=zeros(1,10);
SP1(1:N+1)=0; SP2(1:N+1)=0; SP3(1:N+1)=0; SP4(1:N+1)=0; SP5(1:N+1)=0;
alpha=sqrt(1-rho(2)^2-(rho(5)-rho(1)*rho(2))^2 ...
```

```
    /(1-rho(1)^2));
beta=rho(8)-rho(2)*rho(3)-(rho(5)-rho(1)*rho(2)) ...
    *(rho(6)-rho(1)*rho(3))/(1-rho(1)^2);
delta=1-rho(3)^2-(rho(6)-rho(1)*rho(3))^2/...
    (1-rho(1)^2);
gamma=rho(9)-rho(2)*rho(4)-(rho(5)-rho(1)*rho(2)) ...
    *(rho(7)-rho(1)*rho(4))/(1-rho(1)^2);
lambda=rho(10)-rho(3)*rho(4)-(rho(6)-rho(1)*rho(3)) ...
    *(rho(7)-rho(1)*rho(4))/(1-rho(1)^2);
for j=1:stepn
    payment(1,j)=face*(1+coupon(j));
end
SP1(1)=S(1); SP2(1)=S(2); SP3(1)=S(3); SP4(1)=S(4) ;SP5(1)=S(5); risk=0;
for i=1:ns
    Z=randn(N,5);
    Z1=Z(:,1);
    Z2=rho(1)*Z(:,1)+sqrt(1-rho(1)^2)*Z(:,2);
    Z3=rho(2)*Z(:,1)+(rho(5)-rho(1)*rho(2))/sqrt(1-rho(1)^2) ...
        *Z(:,2)+alpha*Z(:,3);
    Z4=rho(3)*Z(:,1)+(rho(6)-rho(1)*rho(3))/sqrt(1-rho(1)^2) ...
        *Z(:,2)+(beta/alpha)*Z(:,3)+sqrt(delta-(beta/alpha)^2)*Z(:,4);
    Z5=rho(4)*Z(:,1)+(rho(7)-rho(1)*rho(4))/sqrt(1-rho(1)^2) ...
        *Z(:,2)+gamma/alpha*Z(:,3)+(lambda-(beta*gamma/alpha^2)) ...
        /sqrt(delta-(beta/alpha)^2)*Z(:,4)+sqrt(1-rho(4)^2-(rho(7) ...
        -rho(1)*rho(4))^2/(1-rho(1)^2)-(gamma/alpha)^2-(lambda-beta ...
        *gamma/alpha^2)^2/(delta-(beta/alpha)^2))*Z(:,5);

    for j=1:N
        SP1(j+1)=SP1(j)*exp((r-0.5*vol(1)^2)*dt+vol(1)*sqrt(dt)*Z1(j));
        SP2(j+1)=SP2(j)*exp((r-0.5*vol(2)^2)*dt+vol(2)*sqrt(dt)*Z2(j));
        SP3(j+1)=SP3(j)*exp((r-0.5*vol(3)^2)*dt+vol(3)*sqrt(dt)*Z3(j));
        SP4(j+1)=SP4(j)*exp((r-0.5*vol(4)^2)*dt+vol(4)*sqrt(dt)*Z4(j));
        SP5(j+1)=SP5(j)*exp((r-0.5*vol(5)^2)*dt+vol(5)*sqrt(dt)*Z5(j));
    end
    WP=min([SP1; SP2; SP3; SP4; SP5]);
    strike_ch=WP(step);
    payoff(1,:)=0;
    tag=0;
    for j=1:stepn
        if strike_ch(1,j)>=S_rate(j)
            payoff(j)=payment(j);
            tag=1;
            break;
```

```
        end
    end
    if tag == 0
        ki_event = any(WP(:)<kib);
        if ki_event == 0
            if (WP(end)>=kib)
                payoff(1,end)=face*(1+dummy);
            end
        else
            payoff(end)=face*WP(end);
            risk=risk+1;
        end
    end
    dsum=dsum+payoff;
end
exp_payoff=dsum/ns;
for j=1:stepn
    disc_payoff(j)=exp_payoff(j)*exp(-r*step(j)/360);
end
ELS_Price=sum(disc_payoff)
Risk_rate=risk/ns
```