Computational Economics (2025) 65:1147–1168
https://doi.org/10.1007/s10614-024-10590-9

Calibration of Local Volatility Surfaces from Observed Market Call and Put Option Prices

Changwoo Yoo[1,4] · Soobin Kwak[2] · Youngjin Hwang[2] · Hanbyeol Jang[1] · Hyundong Kim[3] · Junseok Kim[2]

Accepted: 11 March 2024 / Published online: 5 April 2024

Abstract

We present a novel, straightforward, robust, and precise calibration algorithm for local volatility surfaces based on observed market call and put option prices. The proposed local volatility reconstruction method is based on the widely recognized generalized Black–Scholes partial differential equation, which is numerically solved using a finite difference scheme. In the proposed method, sample points are strategically placed in the underlying and time domains. The unknown local volatility function is represented using the scattered interpolant function. The primary contribution of this study is that our proposed algorithm not only optimizes the volatility values at the sample points but also optimizes the positions of the sample positions using a least squares method. This optimization process improves the accuracy and robustness of our calibration method. Furthermore, we do not use the Tikhonov regularization technique, which was frequently used to obtain smooth solutions. To validate the practical efficiency and superior performance of the proposed reconstruction method for local volatility functions, we conduct a series of computational experiments using real-world market option prices such as the KOSPI 200, S &P 500, Hang Seng, and Euro Stoxx 50 indices. The proposed algorithm offers financial market practitioners a reliable tool for calibrating local volatility surfaces using only market option prices, enabling more accurate pricing and risk management of financial derivatives.

Keywords Local volatility calibration · Generalized Black–Scholes equation · Call and put options

1 Introduction

We present a novel, straightforward, robust, and precise calibration algorithm for local volatility surfaces based on observed market call and put option prices. A local volatility surface is a mathematical representation of the volatility of an underlying

Extended author information available on the last page of the article asset as a function of both time and the underlying asset. It is a key component in the pricing and risk management of financial derivatives (Deng et al., 2016; Georgiev & Vulkov, 2020; Gong & Xu, 2023; Nabubie & Wang, 2023). The local volatility surface is typically represented as a three-dimensional graph, with time on one axis, the price of the underlying asset on another axis, and volatility on the third axis. It provides a detailed view of how volatility varies across different time and underlying price levels. Calibrating the local volatility surface is an important task in quantitative finance, as it allows for more accurate pricing and risk assessment of complex derivative products. Despite the numerous advantages of the local volatility surface, reconstructing the local volatility surface from market prices poses a significant challenge. Small fluctuations in option prices have a large impact on volatility, making it difficult to reconstruct the local volatility surface using insufficient market price data. Various mathematical and numerical techniques have been developed to model and approximate the local volatility surface based on observed market prices of options. Gong and Xu (2023) developed the process of reconstructing the local volatility surface based on American option prices computed using a finite difference method (FDM), which includes formulating an optimization problem to derive the local volatility by minimizing the disparity between theoretical and market option prices. Nabubie and Wang (2023) considered an inverse problem of determining the time-dependent variable volatility (Kim et al., 2023) from the observed market option prices. Cuomo et al. (2022) presented an algorithm for reconstructing the local volatility surface through multiple procedures using the radial basis function (RBF) method. First, the local volatility surface is reconstructed by evaluating the error resulting from removing points on a globally reconstructed surface. Moreover, local methodologies are employed, including the RBF-partition of unity technique, with variations in subdomain sizes and shape parameters. To price European options and find the volatility surface, they used the proposed model, stochastic differential calculations, and martingale technique.

Georgiev and Vulkov (2021) developed a simple and efficient algorithm that numerically approximates time-dependent implied volatility for jump-diffusion models in option pricing that generalize the Black–Scholes (BS) equation. Kim and Lee (2018) proposed a more effective and efficient algorithm of estimating option prices by combining the local volatility model with the jump-diffusion model. This method estimates the optimal parameter set of the constant volatility jump model and estimates the local volatility surface from the BS implied volatility surface. Wang et al. (2022) used the explicit and Crank–Nicolson method for solving the fractional BS equation. They also used the operator splitting method to implement the semi-implicit method for the multidimensional fractional BS equation. They simulated numerical experiments for option prices, and the numerical results with Greeks behave differently with different Hurst exponents. Cen et al. (2022) applied the adapted posteriori grid method to solve a time-fractional BS equation governing European options. Zhao and Xu (2022) calibrated the time-dependent volatility function for European options. Yan et al. (2022) studied the Heston stochastic volatility model using the explicit finite difference method. They considered American option pricing and proposed transaction costs as small nonlinear price impacts. Ferreiro-Ferreiro et al. (2020) developed a technique for calibrating the whole set of parameters of the Heston stochastic local volatility model in the framework of foreign exchange rate options. This technique involved the Monte–Carlo methods for the Heston stochastic local volatility model and efficient global stochastic optimization algorithms.

The proposed algorithm is based on the widely recognized generalized BS partial differential equation (PDE) (Bustamante & Contreras, 2016; Khodayari & Ranjbar, 2019), an FDM, the least squares method, and observed market call and put option prices. The generalized BS partial differential equation for the European option premium is expressed as follows:

$$\frac{\partial u(S,t)}{\partial t} + \frac{1}{2}[\sigma(S,t)S]^2\frac{\partial^2 u(S,t)}{\partial S^2} + rS\frac{\partial u(S,t)}{\partial S} - ru(S,t) = 0, \tag{1}$$

where r is the risk-neutral interest rate and the variables $u(S,\ t)$ and $\sigma(S,t)$ denote the option value and the local volatility function, respectively. These variables are associated with the underlying asset price S and time t. It is worth noting that Eq. (1) becomes the standard BS PDE if $\sigma(S,t)$ is constant (Lee et al., 2023). In the proposed method, sample points are strategically placed in the underlying and time domains. Our proposed algorithm not only optimizes the volatility values at the sample points but also optimizes the positions of the sample points using the least squares method.

The main purpose of this study is to present a novel, straightforward, robust, and precise calibration algorithm for local volatility surfaces based on observed market call and put option prices so that practitioners can use the proposed method for calibrating local volatility surfaces, enabling more accurate pricing and risk management of financial derivatives.

The paper is organized as follows. In Sect. 2, we propose a methodology for the optimization problem of the local volatility function for call and put options. Furthermore, in Sect. 3, in order to demonstrate the superior performance of the proposed reconstruction algorithm for local volatility functions, we conduct a series of computational experiments using real-world market option prices such as KOSPI 200, S &P 500, Hang Seng, and Euro Stoxx 50 indices. Lastly, we dedicate to discussing the conclusions in Sect. 4.

2 Method

Now, we present a concise overview of the proposed novel method for the optimization problem of the local volatility surface for a European call option. The case of a put option is similarly defined. Let $\tau = T - t$, then we can express Eq. (1) as follows:

$$\frac{\partial u(S,\tau)}{\partial \tau} = \frac{1}{2}[\sigma(S,\tau)S]^2\frac{\partial^2 u(S,\tau)}{\partial S^2} + rS\frac{\partial u(S,\tau)}{\partial S} - ru(S,\tau), \tag{2}$$

which is solved using the FDM on a finite domain $\Omega = [0,\ S_{max}]$. We apply the Dirichlet boundary condition at $S = 0$ and the linear boundary condition at $S = S_{max}$ for the European call option. For the European put option, we apply the linear boundary condition at both $S = 0$ and $S = S_{max}$.

First, we define the underlying domain step size $h = S_{max}/(N_S - 1)$ for some positive integer N_S, and let $S_i = (i - 1)h$ for $i = 1, 2, \ldots, N_S - 1$ (Fig. 1).

For simplicity of notation, let us denote $u(S_i, n\Delta\tau)$ by u_i^n, where $\Delta\tau$ is a time step. Let $\sigma_i^n \equiv \sigma(S_i, n\Delta\tau)$. By applying the implicit Euler scheme (Kim et al., 2021) to Eq. (2), we obtain

$$\frac{u_i^{n+1} - u_i^n}{\Delta\tau} = \frac{(\sigma_i^{n+1}S_i)^2}{2}\left(\frac{u_{i-1}^{n+1} - 2u_i^{n+1} + u_{i+1}^{n+1}}{h^2}\right) + rS_i\left(\frac{u_{i+1}^{n+1} - u_{i-1}^{n+1}}{2h}\right) - ru_i^{n+1}. \tag{3}$$

We can rewrite the above Eq. (3) as

$$\alpha_i u_{i-1}^{n+1} + \beta_i u_i^{n+1} + \gamma_i u_{i+1}^{n+1} = f_i, \quad \text{for} \quad i = 2, \cdots, N_S, \tag{4}$$

where

$$\alpha_i = -\frac{(\sigma_i^{n+1}S_i)^2}{2h^2} + \frac{rS_i}{2h}, \ \beta_i = \frac{1}{\Delta\tau} + \frac{(\sigma_i^{n+1}S_i)^2}{h^2} + r,$$
$$\gamma_i = -\frac{(\sigma_i^{n+1}S_i)^2}{2h^2} - \frac{rS_i}{2h}, \ f_i = \frac{u_i^n}{\Delta\tau}.$$

We solve the tridiagonal system (4) using the Thomas algorithm. In this study, we apply the following boundary conditions for the European call and put options:

$$u_1^{n+1} = 0, \ u_{N_S+1}^{n+1} = 2u_{N_S}^{n+1} - u_{N_S-1}^{n+1}$$

and

$$u_0^{n+1} = 2u_1^{n+1} - u_2^{n+1}, \ u_{N_S+1}^{n+1} = 2u_{N_S}^{n+1} - u_{N_S-1}^{n+1},$$

respectively. Let $\{U_\beta^\alpha\}$ represent the market option prices at time T_α for $\alpha = 1, \ldots, M_\alpha$ and K_β denote the exercise prices for $\beta = 1, \ldots, M_\beta$. Here, $T_1 < \ldots < T_{M_\alpha}$ and $K_1 < \cdots < K_{M_\beta}$. Using $\{U_\beta^\alpha\}$, we calculate $\sigma(S, t)$ by minimizing the following expression (Kim & Kim, 2021):

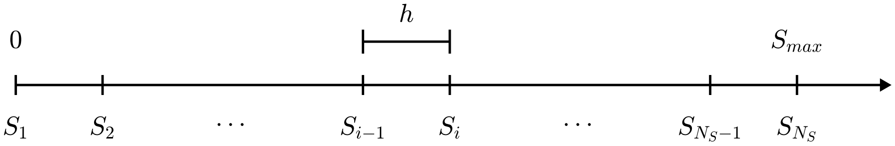


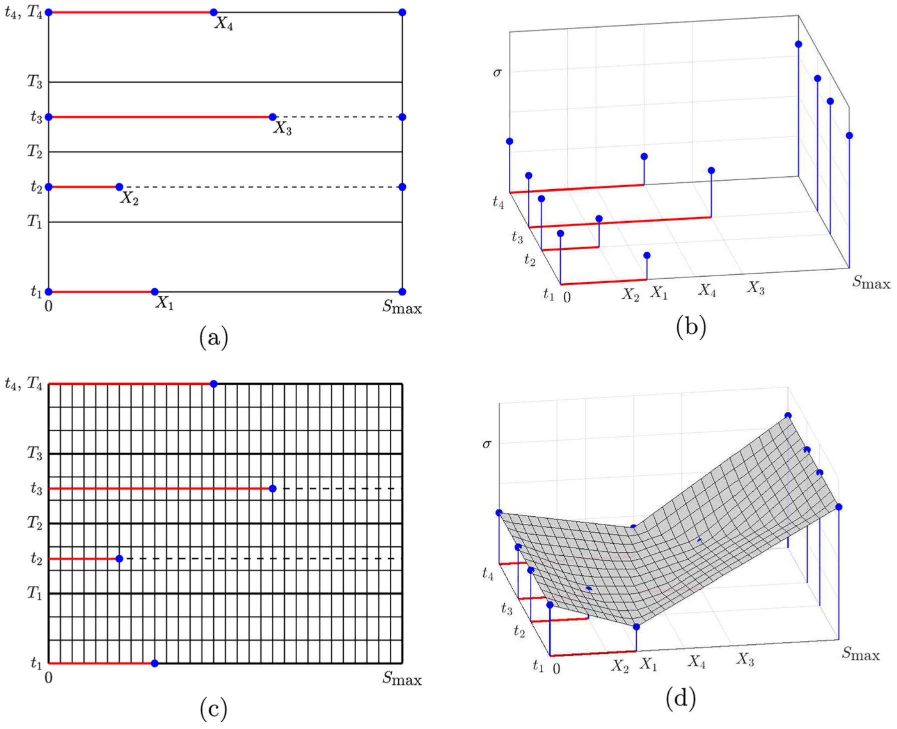
Fig. 1 Discrete underlying domain with an interval size, h

$$\mathcal{E}(\sigma) = \frac{1}{M_\alpha M_\beta} \sum_{\alpha=1}^{M_\alpha} \sum_{\beta=1}^{M_\beta} [u_{K_\beta}(\sigma; S_0, T_\alpha) - U_\beta^\alpha]^2, \tag{5}$$

where $u_{K_\beta}(\sigma; S_0, T_\alpha)$ is the numerical solution of Eq. (2) with $u_i^0 = \max(S_i - K_\beta, 0)$. In this study, we use the *lsqcurvefit* function in MATLAB R2023a (MATLAB, 2021), to calculate the argument minimum σ for $\mathcal{E}(\sigma)$.

In this study, we propose a novel, straightforward, robust, and accurate calibration algorithm for local volatility surfaces based on observed market call and put option prices. The rationale behind our proposed algorithm stems from the lack of uniqueness of local volatility surfaces and the presence of the volatility smile. To prevent the occurrence of highly oscillatory volatility over time, we strategically place sample points at the average times of expiration dates, as demonstrated in (Jin et al., 2018), where the focus was exclusively on reconstructing the time-dependent volatility function using the BS equation. To incorporate the volatility smile and maintain simplicity, we position only three sample points which guarantee the convexity of the volatility function over the underlying asset at each time level.

Specifically, let $t_1 = 0$; $t_q = (T_{q-1} + T_q)/2$ for $q = 2, \ldots, M_\alpha - 1$; and $t_{M_\alpha} = T_{M_\alpha}$ as shown in Fig. 2a. Let us define the volatility values at the sample points as follows:



(a)

(b)

(c)

(d)

Fig. 2 Schematic diagrams of the proposed local volatility surface: Black–Schole sample point positions, **b** volatility values at the sample points, **c** discretization of the computational domain, and **d** interpolated volatility surface

$\sigma(X_q, t_q) = \sigma_q$, $\sigma(0, t_q) = \sigma_{M_\alpha+1}$, and $\sigma(S_{max}, t_q) = \sigma_{M_\alpha+2}$ for $q = 1, \dots, M_\alpha$, where X_q, σ_q, $\sigma_{M_\alpha+1}$, and $\sigma_{M_\alpha+2}$ for $q = 1, \dots, M_\alpha$ are parameters that we need to optimize, see Fig. 2b. Figure 2c, d display the discretization of the computational domain and the interpolated volatility surface, respectively. Here, X_q for $q = 1, \dots, M_\alpha$ represents the position of the sample points. The primary contribution of this study is that our proposed algorithm not only optimizes the volatility values at the sample points but also optimizes the positions of the sample points using a least squares method.

3 Numerical Tests

In this section, we conduct various computational tests to validate the superior performance of the proposed calibration method for local volatility surfaces using observed market call and put option prices.

3.1 Robustness of the Numerical Method

We first test the robustness of the proposed numerical method. It is well known that there is no uniqueness of the volatility surface. To effectively and fairly compare the constructed local volatility surfaces on an effective domain, we consider the following probability density function of a log-normal distribution (Kim & Kim, 2021).

$$f(S,t) = \frac{1}{\sigma S\sqrt{2\pi t}} \exp\left(-\frac{[\ln(S/S_0) - (r - \sigma^2/2)t]^2}{2\sigma^2 t}\right), \; S \in (0, 3S_0), \; t \in (0, 77/365). \tag{6}$$

We use strike prices $K_\beta = 352.5 + 2.5(\beta - 1)$ for $\beta = 1, 2, \cdots, 5$ and maturities $T_1 = 14\Delta\tau$, $T_2 = 42\Delta\tau$, and $T_3 = 77\Delta\tau$, where $\Delta\tau = 1/365$. The present value of the KOSPI 200 index is $S_0 = 357.99$, with an interest rate of $r = 0.0383$, $\sigma = 0.3$, the underlying domain $\Omega = (0, 3S_0)$, and $N_S = 1074$. The premiums for KOSPI 200 index call options at different strikes and maturities are given in Table 1.

The initial parameters are $X_q = S_0$, $\sigma_q = \sigma^*$, $\sigma_{M_\alpha+1} = \sigma^*$, and $\sigma_{M_\alpha+2} = \sigma^*$ for $q = 1, \cdots, M_\alpha$. Here, we test various initial values of σ^* to validate the robustness of the proposed numerical algorithm. Figure 3a shows the reconstructed local volatility surfaces with initial volatility values $\sigma^* = 0.1, 0.2, \cdots, 0.5$ and Fig. 3b displays the reconstructed local volatility surfaces of Fig. 3a on the effective domain. The effective domain is defined as inside the contour of $f(S, t)$ at a level 0.0001 in Eq. (6).

Table 1 Premiums for KOSPI 200 index call options at different strikes and maturities on 28 December 2023

Case	$K_1 = 352.5$	$K_2 = 355$	$K_3 = 357.5$	$K_4 = 360$	$K_5 = 362.5$
$T_1 = 14\Delta\tau$	9.46	7.64	6.13	4.76	3.61
$T_2 = 42\Delta\tau$	12.93	11.35	9.85	8.51	7.29
$T_3 = 77\Delta\tau$	15.78	14.10	12.57	11.20	9.97

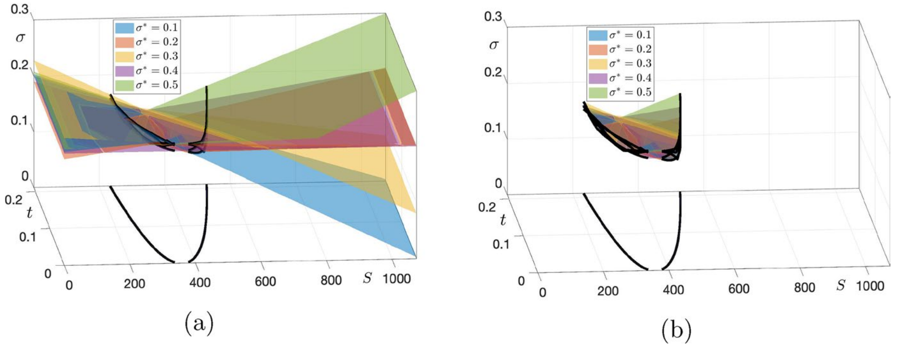


Fig. 3 **a** Reconstructed local volatility surfaces with initial volatility values $\sigma^* = 0.1, 0.2, \cdots, 0.5$ from KOSPI 200 index call options on 28 December 2023. **b** Reconstructed local volatility surfaces of (**a**) on the effective domain

Based on these results, it is evident that the proposed local volatility reconstruction method is robust with respect to the initial guess of the volatility.

3.2 Construction of $\sigma(S, t)$ Using the Market Data from KOSPI 200 Index Options

We test the proposed method by computing a local volatility surface with real market data from KOSPI 200 index call options on 28 December 2023. In this test, we use strike prices $K_\beta = 352.5 + 2.5(\beta - 1)$ for $\beta = 1, 2, \cdots, 5$ and maturities $T_1 = 14\Delta\tau$, $T_2 = 42\Delta\tau$, and $T_3 = 77\Delta\tau$, where $\Delta\tau = 1/365$. The present value of the KOSPI 200 index is $S_0 = 357.99$, and we use an interest rate of $r = 0.0383$. The underlying domain is $\Omega = (0, 3S_0)$ with $N_S = 1074$. The initial parameters are $X_q = S_0$, $\sigma_q = 0.3$, $\sigma_{M_\alpha+1} = 0.3$, and $\sigma_{M_\alpha+2} = 0.3$ for $q = 1, \cdots, M_\alpha$. The premiums for KOSPI 200 index call options at different strikes and maturities are given in Table 1. Fig. 4a shows the local volatility surface reconstructed from the proposed algorithm using real market data from KOSPI 200 index call options. Figure 4b displays the call option market prices (represented by circle symbol) and the computed call option prices (represented by plus symbol) using the reconstructed local volatility surface. We can observe that some market prices deviate from the theoretical prices due to market frictions such as a lack of trading volume and transaction costs. We can also interpret these deviations as either overestimated or underestimated prices compared to the theoretical prices.

Next, we consider the construction of a local volatility surface using real market data from KOSPI 200 index put options. We use strike prices $K_\beta = 352.5 + 2.5(\beta - 1)$ for $\beta = 1, 2, \cdots, 5$, maturities $T_1 = 14\Delta\tau$, $T_2 = 42\Delta\tau$, and $T_3 = 77\Delta\tau$, $S_0 = 357.99$, and $r = 0.0383$. The underlying domain is $\Omega = (0, 3S_0)$ with $N_S = 1074$. The initial parameters are $X_q = S_0$, $\sigma_q = 0.3$, $\sigma_{M_\alpha+1} = 0.3$, and $\sigma_{M_\alpha+2} = 0.3$ for $q = 1, \cdots, M_\alpha$. The premiums for KOSPI 200 index put options at different strikes and maturities are given in Table 2.

Figure 5b displays the reconstructed local volatility surface obtained from the proposed method using the KOSPI 200 index put options. Figure 5c shows the

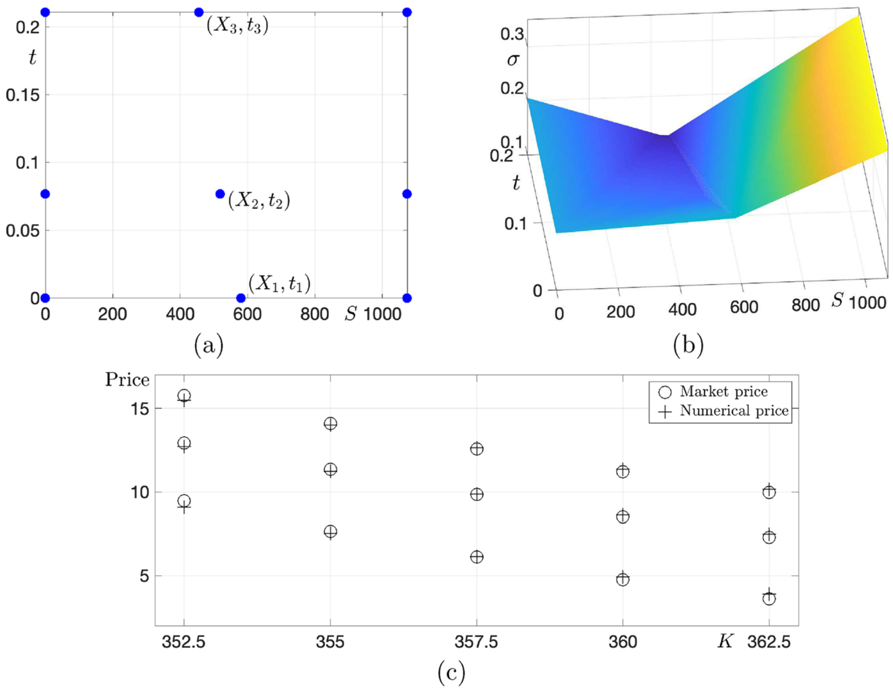


Fig. 4 **a** Positions of reconstructed points. **b** Reconstructed local volatility surface from KOSPI 200 index call options on 28 December 2023. **c** Call option market prices and computed call option prices using the reconstructed local volatility surface

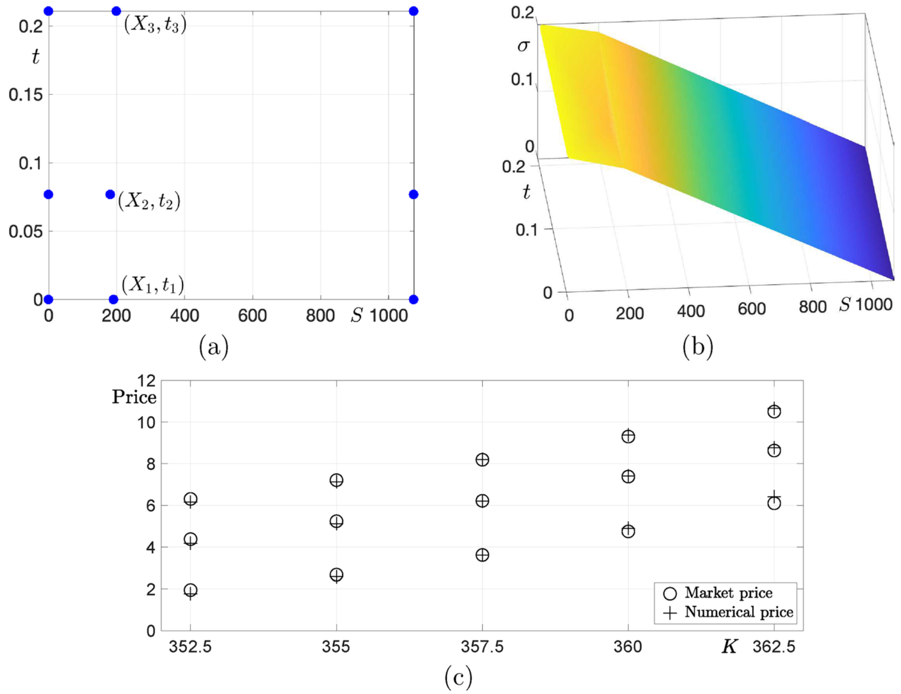
Table 2 Premiums for KOSPI 200 index put options at different strikes and maturities on 28 December 2023

Case	$K_1 = 352.5$	$K_2 = 355$	$K_3 = 357.5$	$K_4 = 360$	$K_5 = 362.5$
$T_1 = 14\Delta\tau$	1.95	2.68	3.62	4.75	6.10
$T_2 = 42\Delta\tau$	4.38	5.24	6.22	7.38	8.63
$T_3 = 77\Delta\tau$	6.31	7.21	8.19	9.30	10.50

market prices of put options (circle symbol) and the computed put option prices (plus symbol) using the reconstructed local volatility surface. It is evident that some market prices deviate from the theoretical prices. We can also interpret these deviations as either overestimated or underestimated prices compared to the theoretical prices.

3.3 Construction of $\sigma(S, t)$ Using the Market Data from S &P 500 Index Options

We apply the proposed approach to construct a local volatility surface using real market data from S &P 500 index call options on 29 December 2023. For this



(a) (b)

(c)

Fig. 5 **a** Reconstructed local volatility surface from KOSPI 200 index put options on 28 December 2023. **b** Positions of reconstructed points. **c** Put option market prices and computed put option prices using the reconstructed local volatility surface

test, we use strike prices $K_\beta = 4760 + 5(\beta - 1)$ for $\beta = 1, 2, \cdots, 5$ and maturities $T_1 = 21\Delta\tau$, $T_2 = 49\Delta\tau$, and $T_3 = 77\Delta\tau$. The current value of the S &P 500 index is $S_0 = 4769.83$, and we use an interest rate of $r = 0.052$. The underlying domain is $\Omega = (0, 3S_0)$ with $N_S = 2862$. The initial parameters are $X_q = S_0$, $\sigma_q = 0.3$, $\sigma_{M_\alpha+1} = 0.3$, and $\sigma_{M_\alpha+2} = 0.3$ for $q = 1, \cdots, M_\alpha$. The premiums for S &P 500 index call options at different strikes and maturities are listed in Table 3.

Figure 6b shows the local volatility surface reconstructed from the proposed algorithm based on the real market data from S &P 500 index call options. Figure 6c displays the market prices of call options (represented by circle symbol) and the computed call option prices (represented by plus symbol) using the

Table 3 Premiums for S &P 500 index call options at different strikes and maturities on 29 December 2023

Case	$K_1 = 4760$	$K_2 = 4765$	$K_3 = 4770$	$K_4 = 4775$	$K_5 = 4780$
$T_1 = 21\Delta\tau$	58.20	55.10	52.10	49.20	46.55
$T_2 = 49\Delta\tau$	101.30	98.15	94.95	91.85	88.80
$T_3 = 77\Delta\tau$	135.35	132.05	128.80	125.60	122.40

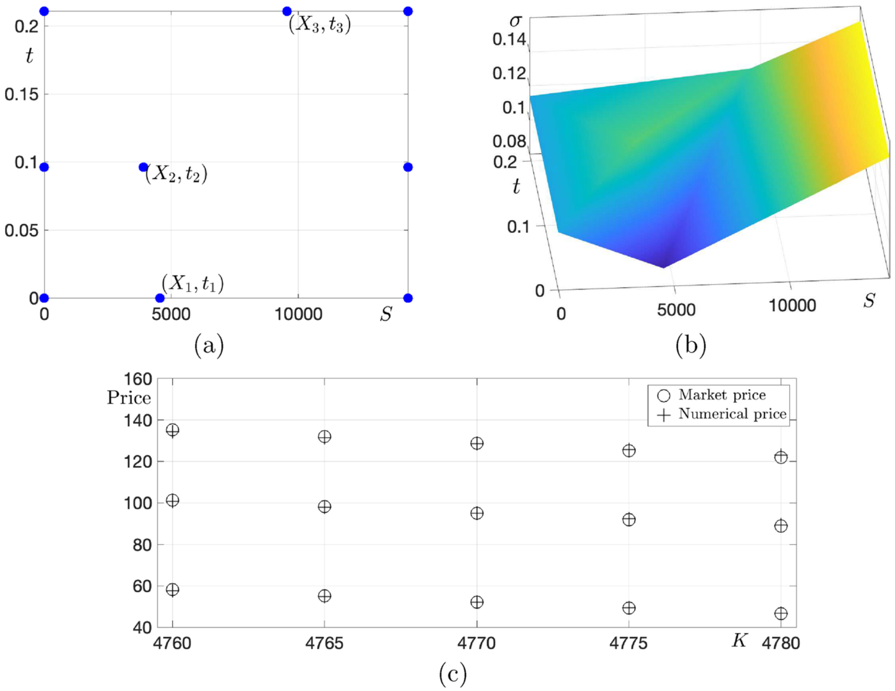


Fig. 6 **a** Reconstructed local volatility surface from S &P 500 index call options on 29 December 2023. **b** Positions of reconstructed points. **c** Call option market prices and computed call option prices using the reconstructed local volatility surface

reconstructed local volatility surface. We can observe that some market prices deviate from the theoretical prices due to market frictions such as limited trading volume and transaction costs. These deviations can be interpreted as either overestimated or underestimated prices compared to the theoretical prices.

Next, we consider the construction of a local volatility surface using real market data from S &P 500 index put options. We use strike prices $K_\beta = 4760 + 5(\beta - 1)$ for $\beta = 1, 2, \cdots, 5$, maturities $T_1 = 21\Delta\tau$, $T_2 = 49\Delta\tau$, and $T_3 = 77\Delta\tau$, $S_0 = 4769.83$, and $r = 0.052$. The underlying domain $\Omega = (0, 3S_0)$ and $N_S = 2862$. The initial parameters are $X_q = S_0$, $\sigma_q = 0.3$, $\sigma_{M_\alpha+1} = 0.3$, and $\sigma_{M_\alpha+2} = 0.3$ for $q = 1, \cdots, M_\alpha$. The premiums for S &P 500 index put options at different strikes and maturities can be found in Table 4.

Figure 7b displays the reconstructed local volatility surface obtained from the proposed algorithm using S &P 500 index put options. Figure 7c shows the market prices of put options (circle symbol) and the computed put option prices (plus symbol) using the reconstructed local volatility surface. It is evident that some market prices deviate from the theoretical prices. We can also interpret these deviations as either overestimated or underestimated prices compared to the theoretical prices.

Table 4 Premiums for S &P 500 index put options at different strikes and maturities on 29 December 2023

Case	$K_1 = 4760$	$K_2 = 4765$	$K_3 = 4770$	$K_4 = 4775$	$K_5 = 4780$
$T_1 = 21\Delta\tau$	35.35	37.20	39.20	41.30	43.45
$T_2 = 49\Delta\tau$	62.80	64.55	66.40	68.20	70.10
$T_3 = 77\Delta\tau$	82.55	84.15	85.85	87.60	89.35

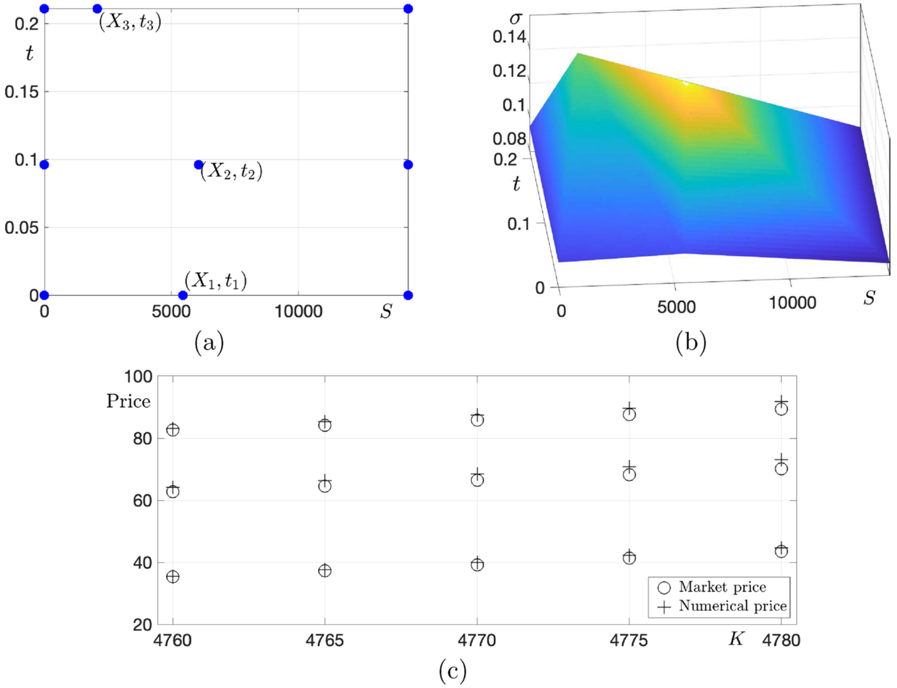


Fig. 7 **a** Reconstructed local volatility surface from S &P 500 index put options on 29 December 2023. **b** Positions of reconstructed points. **c** Put option market prices and computed put option prices using the reconstructed local volatility surface

3.4 Construction of $\sigma(S, t)$ Using the Market Data from Hang Seng Index Options

We conduct a numerical simulation to validate the performance of the proposed method by computing a local volatility surface using real market data from Hang Seng index call options on 29 December 2023. We use strike prices $K_\beta = 16800 + 100(\beta - 1)$ for $\beta = 1, 2, \cdots, 5$ and maturities $T_1 = 32\Delta\tau$, $T_2 = 61\Delta\tau$, and $T_3 = 89\Delta\tau$, where $\Delta\tau = 1/365$. The present value of the Hang Seng index is set at $S_0 = 17047.39$, and we use an interest rate of $r = 0.0446$. The underlying domain is $\Omega = (0, 3S_0)$ with $N_S = 5114$. The initial parameters are $X_q = S_0$,

Table 5 Premiums for Hang Seng index call options at different strikes and maturities on 29 December 2023

Case	$K_1 = 16800$	$K_2 = 16900$	$K_3 = 17000$	$K_4 = 17100$	$K_5 = 17200$
$T_1 = 32\Delta\tau$	601	541	485	433	386
$T_2 = 61\Delta\tau$	811	751	696	643	592
$T_3 = 89\Delta\tau$	942	885	831	778	727

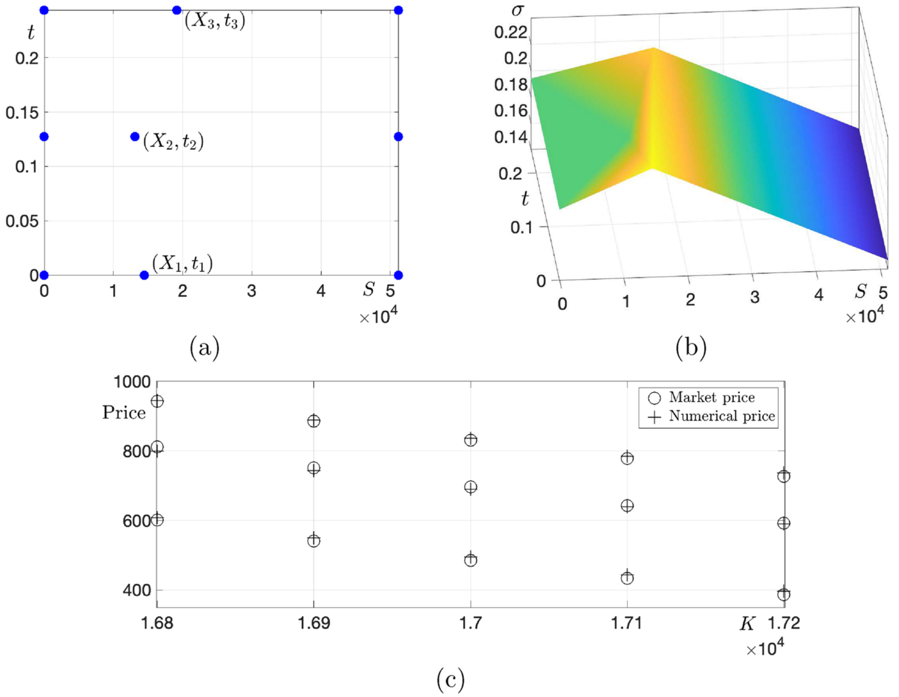


Fig. 8 **a** Reconstructed local volatility surface from Hang Seng index call options on 29 December 2023. **b** Positions of reconstructed points. **c** Call option market prices and computed call option prices using the reconstructed local volatility surface

$\sigma_q = 0.3$, $\sigma_{M_\alpha+1} = 0.3$, and $\sigma_{M_\alpha+2} = 0.3$ for $q = 1, \cdots, M_\alpha$. The premiums for call options at different strikes and maturities are listed in Table 5.

Figure 8b shows the local volatility surface reconstructed using the proposed algorithm with real market data from Hang Seng index call options. Figure 8c displays the Hang Seng index call option market prices (represented by circle symbol) and the computed Hang Seng index call option prices (represented by plus symbol) using the reconstructed local volatility surface. We can observe that some market prices deviate from the theoretical prices due to market frictions such as a lack of trading volume and transaction costs. These deviations can also be interpreted as either overestimated or underestimated prices compared to the theoretical prices.

Table 6 Premiums for Hang Seng index put options at different strikes and maturities on 29 December 2023

Case	$K_1 = 16800$	$K_2 = 16900$	$K_3 = 17000$	$K_4 = 17100$	$K_5 = 17200$
$T_1 = 32\Delta\tau$	273	313	357	401	459
$T_2 = 61\Delta\tau$	409	450	493	539	589
$T_3 = 89\Delta\tau$	550	593	638	685	735

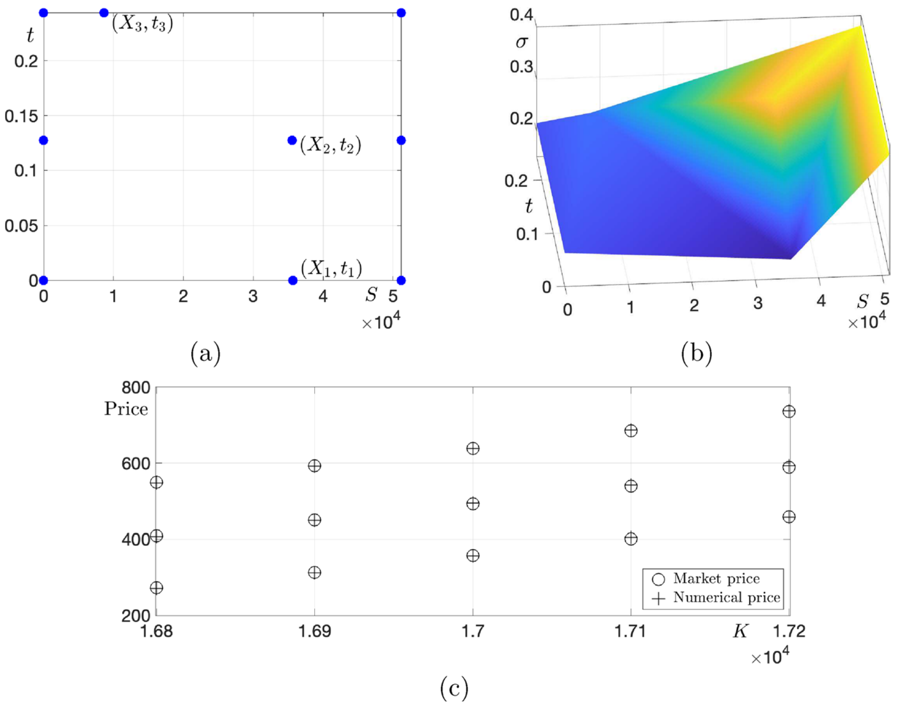


Fig. 9 **a** Reconstructed local volatility surface from Hang Seng index put options on 29 December 2023. **b** Positions of reconstructed points. **c** Put option market prices and computed put option prices using the reconstructed local volatility surface

Next, we consider the creation of a local volatility surface using actual market data from Hang Seng index put options on 10 July 2023. We use strike prices $K_\beta = 16800 + 100(\beta - 1)$ for $\beta = 1, 2, \cdots, 5$, maturities $T_1 = 32\Delta\tau$, $T_2 = 61\Delta\tau$, and $T_3 = 89\Delta\tau$, where $\Delta\tau = 1/365$, $S_0 = 17047.39$, and $r = 0.0446$. The underlying domain is $\Omega = (0, 3S_0)$ with $N_S = 5114$. The initial parameters are $X_q = S_0$, $\sigma_q = 0.3$, $\sigma_{M_\alpha+1} = 0.3$, and $\sigma_{M_\alpha+2} = 0.3$ for $q = 1, \cdots, M_\alpha$. The premiums for Hang Seng index put options at different strikes and maturities are given in Table 6.

Figure 9b displays the reconstructed local volatility surface obtained from the proposed method using Hang Seng index put options. Figure 9c shows the market prices of Hang Seng index put options (circle symbol) and the computed prices of Hang Seng index put options (plus symbol) using the reconstructed local volatility surface. Clearly, there are discrepancies between some market prices and the theoretical prices. These disparities can be interpreted as either overestimated or underestimated prices compared to the theoretical prices.

3.5 Construction of $\sigma(S, t)$ Using the Market Data from Euro Stoxx 50 Index Options

We compute a local volatility surface with real market data from Euro Stoxx 50 index call options on 29 December 2023. In this test, we use $N_S = 1356$, strike prices $K_\beta = 4475 + 25(\beta - 1)$ for $\beta = 1, 2, \cdots, 5$, and maturities $T_1 = 21\Delta\tau$, $T_2 = 49\Delta\tau$, and $T_3 = 77\Delta\tau$, where $\Delta\tau = 1/365$. The present value of the Euro Stoxx 50 index is $S_0 = 4521.65$, and we use an interest rate of $r = 0.03909$. The underlying domain $\Omega = (0, 3S_0)$. The initial parameters are $X_q = S_0$, $\sigma_q = 0.3$, $\sigma_{M_\alpha+1} = 0.3$, and $\sigma_{M_\alpha+2} = 0.3$ for $q = 1, \cdots, M_\alpha$. The premiums for Euro Stoxx 50 index call options at different strikes and maturities are listed in Table 7.

Figure 10b shows the local volatility surface reconstructed from the proposed algorithm using real market data from Euro Stoxx 50 index call options. Figure 10c displays call option market prices (circle symbol) and computed call option prices (plus symbol) using the reconstructed local volatility surface. We can observe that when the strike price is $K = 4475$, the market prices are lower than the theoretical values for a call option, indicating that the option price is undervalued. On the other hand, when $K = 4575$ and $T = 77\Delta\tau$, the market price is higher than the theoretical value. Thus, we can conclude that the option price is overvalued.

Next, we consider the construction of a local volatility surface using real market data from Euro Stoxx 50 index put options. We use strike prices $K_\beta = 4475 + 25(\beta - 1)$ for $\beta = 1, 2, \cdots, 5$, maturities $T_1 = 21\Delta\tau$, $T_2 = 49\Delta\tau$, and $T_3 = 77\Delta\tau$, where $\Delta\tau = 1/365$, $S_0 = 4521.65$, and $r = 0.03909$. The underlying domain $\Omega = (0, 3S_0)$ and $N_S = 1356$. The initial parameters are $X_q = S_0$, $\sigma_q = 0.3$, $\sigma_{M_\alpha+1} = 0.3$, and $\sigma_{M_\alpha+2} = 0.3$ for $q = 1, \cdots, M_\alpha$. The premiums for Euro Stoxx 50 index put options at different strikes and maturities are given in Table 8.

Figure 11b illustrates the reconstructed local volatility surface obtained from the proposed method using Euro Stoxx 50 index put options. Figure 11c displays Euro Stoxx 50 put option market prices (circle symbol) and computed Euro Stoxx 50 put option prices (plus symbol) using the reconstructed local volatility surface. It

Table 7 Premiums for Euro Stoxx 50 index call options at different strikes and maturities on 29 December 2023

Case	$K_1 = 4475$	$K_2 = 4500$	$K_3 = 4525$	$K_4 = 4550$	$K_5 = 4575$
$T_1 = 21\Delta\tau$	78.4	61.9	47.5	35.3	25.4
$T_2 = 49\Delta\tau$	114.8	98.8	84.1	70.6	58.6
$T_3 = 77\Delta\tau$	149.1	132.8	117.6	103.3	90.1

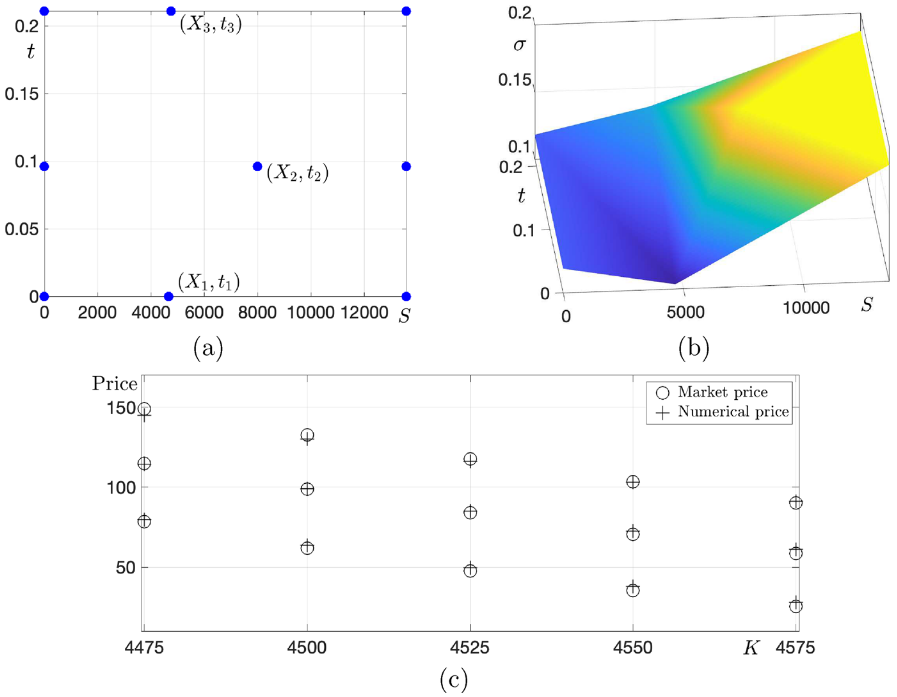


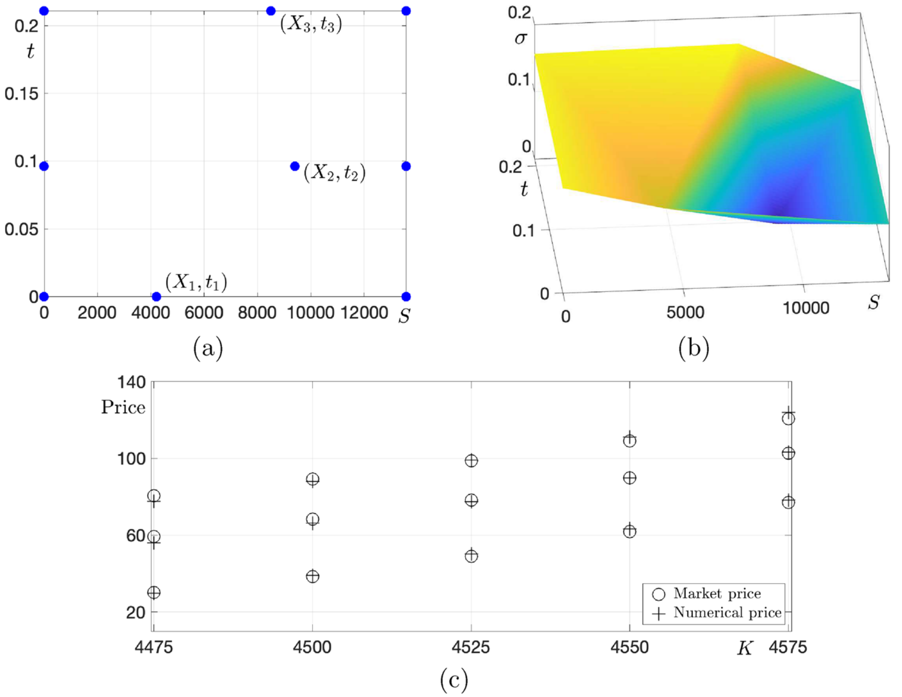
Fig. 10 **a** Reconstructed local volatility surface from Euro Stoxx 50 index call options on 29 December 2023. **b** Positions of reconstructed points. **c** Call option market prices and computed call option prices using the reconstructed local volatility surface

Table 8 Premiums for Euro Stoxx 50 index put options at different strikes and maturities on 29 December 2023

Case	$K_1 = 4475$	$K_2 = 4500$	$K_3 = 4525$	$K_4 = 4550$	$K_5 = 4575$
$T_1 = 21\Delta\tau$	30.1	38.5	49.0	61.8	76.9
$T_2 = 49\Delta\tau$	59.3	68.2	78.3	89.8	102.6
$T_3 = 77\Delta\tau$	80.5	89.1	98.6	109.1	120.6

is evident that some market prices deviate from the theoretical prices. We can also interpret these deviations as either overestimated or underestimated prices compared to the theoretical prices.

Next, let us now turn our attention to the comparison of the volatility surfaces for the four different stock indexes. We consider the local volatility surfaces reconstructed from the four different stock indices. This comparative test provides valuable insights into their respective behavior patterns and dynamic characteristics exhibited by each index and helps our understanding of their respective dynamics. We scale the underlying asset price for each stock index to compare different stock indices as $\bar{S} = S/S_0$, for $S \in (0, 3S_0)$. Figure 12a and b show the reconstructed local

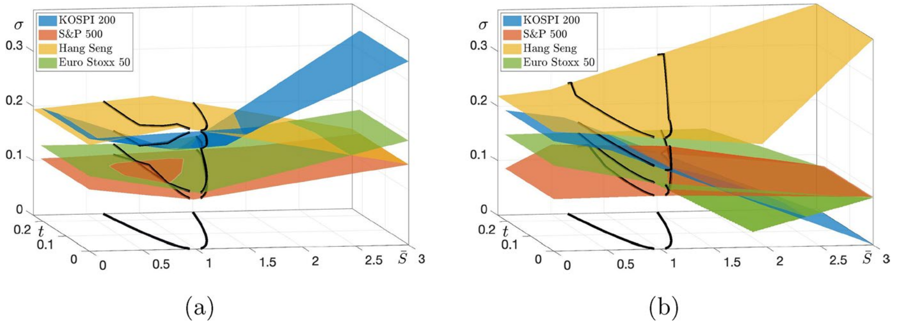


(a)

(b)

(c)

Fig. 11 **a** Reconstructed local volatility surface from Euro Stoxx 50 index put options on 29 December 2023. **b** Positions of reconstructed points. **c** Put option market prices and computed put option prices using the reconstructed local volatility surface



(a)

(b)

Fig. 12 Reconstructed local volatility surfaces of the call **a** and put **b** options on the four different stock indices

volatility surfaces $\sigma(\bar{S}, t)$ of the call and put options on stock indices with a contour of the probability density function, respectively. Here, we use the $\sigma = 0.18$, $r = 0.03$, and $S_0 = 1$ for probability density function. The effective domain is defined as the inside the contours of $f(\bar{S}, t)$ at a level 0.0001 in Eq. (6). We observe variations in the local volatility surfaces among the different stock indices, with some displaying higher values while others exhibit lower levels compared to their counterparts.

Table 9 Premiums for KOSPI 200 index call options at different strikes and maturities on 4 October 2023

Case	$K_1 = 315$	$K_2 = 317.5$	$K_3 = 320$	$K_4 = 322.5$	$K_5 = 325$
$T_1 = 8\Delta\tau$	5.73	4.18	2.73	1.67	0.93
$T_2 = 36\Delta\tau$	9.10	7.90	6.60	5.28	4.04
$T_3 = 64\Delta\tau$	12.40	10.85	9.45	8.39	7.13

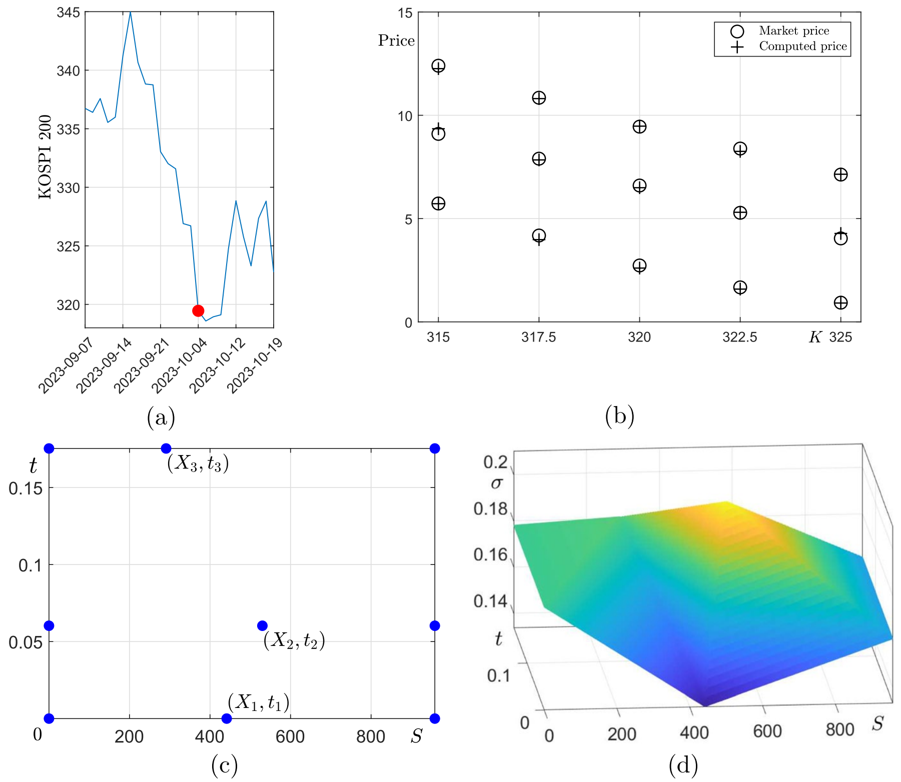


Fig. 13 **a** KOSPI 200 index. **b** Call option market prices and computed call option prices using the reconstructed local volatility surface. **c** Positions of reconstructed points. **d** Reconstructed local volatility surface from KOSPI 200 index call options on 04 October 2023

3.6 Sharp Fluctuation of the Stock Index

We perform a numerical simulation demonstrating the capability of the proposed volatility surface calibration algorithm in handling sharp fluctuations of the KOSPI 200 index. For the numerical experiment, we use call option price data from the day of sharp fluctuations in the KOSPI 200 index, as indicated by the red dot in Fig. 13a. The call option prices for the KOSPI 200 index are listed in Table 9.

Table 10 Premiums for KOSPI 200 index call options at different strikes and maturities on 4 January 2024

Case	$K_1 = 352.5$	$K_2 = 355$	$K_3 = 357.5$	$K_4 = 360$	$K_5 = 362.5$
$T_1 = 8\Delta\tau$	1.98	1.35	0.92	0.61	0.40
$T_2 = 36\Delta\tau$	5.82	4.79	3.25	3.25	2.60
$T_3 = 64\Delta\tau$	10.00	7.48	5.71	5.71	4.69

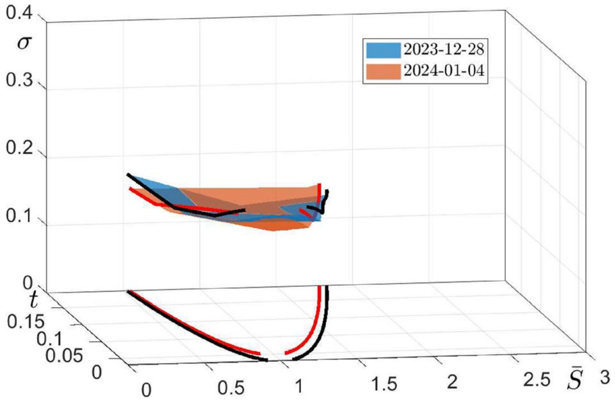


Fig. 14 Reconstructed local volatility surfaces of the call options on 28 December 2023 and 4 January 2024

Figure 13a–d show the KOSPI 200 index, call option market prices, computed prices by the proposed method, positions of reconstructed points, and reconstructed local volatility function using the KOSPI 200 index call option prices, respectively. In this test, we used the same model and numerical parameter values as in the previous case of the KOSPI 200 index. The computational results indicate that the proposed volatility surface calibration can effectively handle relatively sharp changes in the underlying index.

3.7 Historical Market Data

In this section, we consider the construction of local volatility surfaces using two sets of historical market data and conduct a comparative analysis. The primary objective of this test is to explore the predictive capability of the proposed model. For the numerical simulation, we use the KOSPI 200 index call option prices on 28 December 2023 and 4 January 2024. The call option prices for the KOSPI 200 index on 28 December 2023 are identical to the data used in Sec. 3.2. The call option prices for the KOSPI 200 index on 4 January 2024 are listed in Table 10.

Figure 14 shows the reconstructed local volatility surfaces on the effective domains of the call options on 28 December 2023 and 4 January 2024. Here, we used strike prices $K_\beta = 352.5 + 2.5(\beta - 1)$ for $\beta = 1, 2, \cdots, 5$, $\Delta\tau = 1/365$, $\sigma = 0.3$, an interest rate of $r = 0.0383$, and the underlying domain $\Omega = (0, 3S_0)$. The KOSPI 200 indices on 28 December 2023 and 4 January 2024 are 357.99 and 348.07, respectively.

The local volatility surface on 4 January 2024 is slightly tilted compared to that on 28 December 2023. However, the overall shape is similar to each other in the common region. Based on this observation, we can predict the future behavior of the local volatility surface from its current state.

4 Conclusions

In this paper, we proposed a simple, robust, and efficient computational algorithm for reconstructing the local volatility surface using the generalized BS equation. We reconstructed the local volatility function that best fits the theoretical and market option prices. We conducted a computational experiment on the reconstruction of the local volatility functions for the KOSPI 200 index, S &P 500 index, Hang Seng index, and Euro Stoxx 50 index options. The numerical experiments validated the high performance of the proposed algorithm for reconstructing the local volatility surface. In addition, we will extend the proposed method in the future to reconstruct the local volatility function for Bitcoin, one of the most volatile cryptocurrencies as a virtual currency (Azizi et al., 2023). Volatility plays a crucial role in financial economics and investment theory. Scholars in this area study volatility to understand the risk and return profiles of financial assets. They analyze volatility patterns to make informed investment decisions, construct portfolio strategies, and develop pricing models for options and derivatives. Hence, the analysis of volatility provides a prominent position in contemporary finance and drives innovation and influencing the development of investment strategies and financial decision-making frameworks.

5 Use of AI Tools Declaration

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

Appendix

The following MATLAB source code is also available from the corresponding author's webpage: https://mathematicians.korea.ac.kr/cfdkim/open-source-codes/

```
clear all; clf;
global Nx h x r dt S0 t T T2 NK_call xx_call yy_call Xq Yq A_call
% KOSPI200 Call Option Price on 30 December 2021
market_call=[6.80 4.03 2.13 1.03 0.47; 10.25 7.50 5.13 3.45 2.20
             12.80 10.50 8.05 6.65 4.15];
T(1)=datenum(2022,01,13)-datenum(2021,12,30);
T(2)=datenum(2022,02,10)-datenum(2021,12,30);
T(3)=datenum(2022,03,10)-datenum(2021,12,30);
LT=length(T); dt=1/365; t=linspace(0,dt*T(LT),T(LT)+1); S0=394.19; r=0.02;
Lx=3*S0; Nx=round(Lx); x=linspace(0,Lx,Nx); h=x(2)-x(1);
[Xq,Yq] = meshgrid(x,t); T2(1)=0;
for i=2:LT-1
    T2(i)=(T(i-1)+T(i))/2;
end
T2(LT)=T(LT); ST_call=[390:5:410]; NK_call=length(ST_call);
for k=1:LT
    for i=1:NK_call
        A_call(NK_call*(k-1)+i,1)=ST_call(i);
        A_call(NK_call*(k-1)+i,2)=T(k);
        exv_call(NK_call*(k-1)+i)=market_call(k,i);
    end
end
xdata_call=A_call(:,1); CV_call=exv_call'; % market_call Price
vol0_call=ones(8,1)*0.3; lb_call=0*vol0_call; ub_call=0*vol0_call+1;
vol0_call(end-2:end)=1; ub_call(end-2:end)=3;
xx_call=[0 0 0 Lx Lx Lx vol0_call(end-2)*S0 vol0_call(end-1)*S0 ...
    vol0_call(end)*S0]';
yy_call=[dt*T2 dt*T2 dt*T2]';
figure(1); clf; plot(xx_call,yy_call,'o'); grid on; hold on
w_call=[vol0_call(1)*ones(3,1); vol0_call(2)*ones(3,1); vol0_call(3:5)];
F_call = scatteredInterpolant(xx_call,yy_call,w_call);
for i=1:size(Xq,2)
   for j=1:size(Xq,1)
        vf_call(i,j)=F_call(x(i),t(j));
   end
end
max_it=5; options = optimset('MaxIter',max_it);
vol_call = lsqcurvefit(@BScallT, vol0_call, xdata_call, CV_call, ...
    lb_call,ub_call, options);
w_call=[vol_call(1)*ones(3,1); vol_call(2)*ones(3,1); vol_call(3:5)];
xx_call=[0 0 0 x(end) x(end) x(end) vol_call(end-2)*S0 vol_call(end-1)*S0 ...
    vol_call(end)*S0]';
F_call = scatteredInterpolant(xx_call,yy_call,w_call);
for i=1:size(Xq,2)
   for j=1:size(Xq,1)
       vf_call(i,j)=F_call(x(i),t(j));
   end
end
plot(xx_call,yy_call,'b*'); grid on; hold on
xx_call1=xx_call; fs = 17;
figure(3); clf; hold on; grid on; set(gca,'fontsize',15)
mesh(x,t,vf_call'); axis([0 1200 0 0.2 0.1 0.25]); view([-5 45])
text('Interpreter','latex','String','$S$','position',[1080 -0.03 ...
    0.1],'fontsize',fs)
text('Interpreter','latex','String','$t$','position',[-108 0.0785 ...
    0.168],'fontsize',fs)
lw=1; fs=18; ms=10; ph=3; height=15;
figure(7); clf; hold on; box on; grid on
V_call=BScallT(vol_call, xdata_call);
plot(A_call(:,1),CV_call,'ko','linewidth',lw,'markersize',ms);
plot(A_call(:,1),V_call,'k+','linewidth',lw,'markersize',ms);
axis([ST_call(1)-.5 ST_call(end)+.5 -0.05 height]); set(gca,'fontsize',fs)
xticks(ST_call)
legend('Interpreter','latex','string',{'Market price','Numerical price', ...
    'Numerical Call Put'}, ...
    'location','northeast')
text('Interpreter','latex','string','$K$','fontsize',fs, 'position',[407.7 ...
    -1.11265 0])
text('Interpreter','latex','string','Price','fontsize',fs, 'position',[386.5 ...
    13.77 0])

function V=BScallT(vol_call, xdata_call)
 global Nx h x t r dt S0 xx_call yy_call A_call;
 w_call=[vol_call(1)*ones(3,1); vol_call(2)*ones(3,1); vol_call(3:5)];
 xx_call=[0 0 0 x(end) x(end) x(end) vol_call(end-2)*S0 vol_call(end-1)*S0 ...
     vol_call(end)*S0]';
 F_call = scatteredInterpolant(xx_call,yy_call,w_call);
 for i=1:Nx
    for j=1:length(t)
        vf_call(i,j)=F_call(x(i),t(j));
    end
 end
for j=1:length(A_call(:,1))
  Nt=A_call(j,2); u(:,1)=max(x-A_call(j,1),0);
  for n=1:Nt
   for i=2:Nx
    a(i-1)=r*x(i)/(2*h)-(vf_call(i,Nt-n+1)*x(i))^2/(2*h^2);
    d(i-1)=1/dt+(vf_call(i,Nt-n+1)*x(i))^2/h^2+r;
    c(i-1)=-r*x(i)/(2*h)-(vf_call(i,Nt-n+1)*x(i))^2/(2*h^2);
   end
    a(Nx-1)=a(Nx-1)-c(Nx-1); d(Nx-1)=d(Nx-1)+2*c(Nx-1);
    b=u(2:Nx,n)/dt; u(2:Nx,n+1)=thomas(a,d,c,b);
  end
    V(j,1)=interp1(x,u(:, Nt+1),S0);
end
end

function x = thomas(alpha,beta,gamma,f)
n = length(f);
for  i = 2:n
    mult = alpha(i)/beta(i-1); beta(i) = beta(i)-mult*gamma(i-1);
    f(i) = f(i)-mult*f(i-1);
end
x(n) = f(n)/beta(n);
for i = n-1:-1:1
    x(i) = (f(i)-gamma(i)*x(i+1))/beta(i);
end
end
```

Acknowledgements Hyundong Kim was supported by the Research Institute of Natural Science of Gangneung-Wonju National University. This study was supported by 2023 Academic Research Support Program in Gangneung-Wonju National University. The corresponding author (J.S. Kim) was supported by the Brain Korea 21 FOUR through the National Research Foundation of Korea funded by the Ministry of Education of Korea. The authors would like to express their gratitude to the reviewers for the valuable feedback on this paper.

Funding The authors have not disclosed any funding.

Declarations

Conflict of interest The authors declare there is no conflicts of interest.

References

Azizi, S. P., Huang, C. Y., Chen, T. A., Chen, S. C., & Nafei, A. (2023). Bitcoin volatility forecasting: An artificial differential equation neural network. *AIMS Mathematics, 8*(6), 13907–13922. https://doi.org/10.3934/math.2023712

Bustamante, M., & Contreras, M. (2016). Multi-asset Black-Scholes model as a variable second class constrained dynamical system. *Physica A: Statistical Mechanics and its Applications, 457*, 540–572. https://doi.org/10.1016/j.physa.2016.03.063

Cen, Z., Huang, J., & Xu, A. (2022). A posteriori grid method for a time-fractional Black–Scholes equation. *AIMS Mathematics, 7*(12), 20962–20978. https://doi.org/10.3934/math.20221148

Cuomo, S., De Rossi, A., Rizzo, L., & Sica, F. (2022). Reconstruction of volatility surfaces: A first computational study. *Dolomites Research Notes on Approximation*. https://doi.org/10.14658/PUPJ-DRNA-2022-3-5

Deng, Z. C., Hon, Y. C., & Isakov, V. (2016). Recovery of time-dependent volatility in option pricing model. *Inverse Problems, 32*(11), 115010. https://doi.org/10.1088/0266-5611/32/11/115010

Ferreiro-Ferreiro, A. M., García-Rodríguez, J. A., Souto, L., & Vázquez, C. (2020). A new calibration of the Heston stochastic local volatility model and its parallel implementation on GPUs. *Mathematics and Computers in Simulation, 177*, 467–486. https://doi.org/10.1016/j.matcom.2020.04.001

Georgiev, S. G., & Vulkov, L. G. (2020). Computational recovery of time-dependent volatility from integral observations in option pricing. *Journal of Computational Science, 39*, 101054. https://doi.org/10.1016/j.jocs.2019.101054

Georgiev, S. G., & Vulkov, L. G. (2021). Computation of the unknown volatility from integral option price observations in jump-diffusion models. *Mathematics and Computers in Simulation, 188*, 591–608. https://doi.org/10.1016/j.matcom.2021.05.008

Gong, W., & Xu, Z. (2023). Reconstruction of local volatility surface from American options. *Journal of Inverse and Ill-posed Problems, 31*(1), 91–102. https://doi.org/10.1515/jiip-2019-0085

Jin, Y., Wang, J., Kim, S., Heo, Y., Yoo, C., Kim, Y., Kim, J., & Jeong, D. (2018). Reconstruction of the time-dependent volatility function using the Black–Scholes model. *Discrete Dynamics in Nature and Society*. https://doi.org/10.1155/2018/3093708

Khodayari, L., & Ranjbar, M. (2019). A computationally efficient numerical approach for multi-asset option pricing. *International Journal of Computer Mathematics, 96*(6), 1158–1168. https://doi.org/10.1080/00207160.2018.1458096

Kim, S., Han, H., Jang, H., Jeong, D., Lee, C., Lee, W., & Kim, J. (2021). Reconstruction of the local volatility function using the Black-Scholes model. *Journal of Computational Science, 51*, 101341. https://doi.org/10.1016/j.jocs.2021.101341

Kim, S., & Kim, J. (2021). Robust and accurate construction of the local volatility surface using the Black–Scholes equation. *Chaos, Solitons & Fractals, 150*, 111116. https://doi.org/10.1016/j.chaos.2021.111116

Kim, N., & Lee, Y. (2018). Estimation and prediction under local volatility jump-diffusion model. *Physica A: Statistical Mechanics and its Applications, 491*, 729–740. https://doi.org/10.1016/j.physa.2017.09.035

Kim, S., Lyu, J., Lee, W., Park, E., Jang, H., Lee, C., & Kim, J. (2023). A Practical Monte Carlo Method for Pricing Equity-Linked Securities with Time-Dependent Volatility and Interest Rate. *Computational Economics,1–18*,. https://doi.org/10.1007/s10614-023-10394-3

Lee, C., Kwak, S., Hwang, Y., & Kim, J. (2023). Accurate and efficient finite difference method for the Black–Scholes model with no far-field boundary conditions. *Computational Economics, 61*(3), 1207–1224. https://doi.org/10.1007/s10614-022-10242-w

MATLAB, (2021) Version 9.10. 0.1602886 (R2021a), The MathWorks Inc., Natick, Massachusetts.

Nabubie, B., & Wang, S. (2023). Numerical techniques for determining implied volatility in option pricing. *Journal of Computational and Applied Mathematics, 422*, 114913. https://doi.org/10.1016/j.cam.2022.114913

Wang, J., Wen, S., Yang, M., & Shao, W. (2022). Practical finite difference method for solving multi-dimensional Black–Scholes model in fractal market. *Chaos, Solitons & Fractals, 157*, 111895. https://doi.org/10.1016/j.chaos.2022.111895

Yan, D., Lin, S., Hu, Z., & Yang, B. Z. (2022). Pricing American options with stochastic volatility and small nonlinear price impact: A PDE approach. *Chaos, Solitons & Fractals, 163*, 112581. https://doi.org/10.1016/j.chaos.2022.112581

Zhao, J. J., & Xu, Z. L. (2022). Calibration of time-dependent volatility for European options under the fractional Vasicek model. *AIMS Math, 7*, 11053–11069. https://doi.org/10.3934/math.2022617

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Changwoo Yoo[1,4] · Soobin Kwak[2] · Youngjin Hwang[2] · Hanbyeol Jang[1] · Hyundong Kim[3] · Junseok Kim[2]

✉ Junseok Kim
cfdkim@korea.ac.kr

Changwoo Yoo
coreapoa@gmail.com

Soobin Kwak
soobin23@korea.ac.kr

Youngjin Hwang
youngjin_hwang@korea.ac.kr

Hanbyeol Jang
styliststar@korea.ac.kr

Hyundong Kim
hdkim@gwnu.ac.kr

[1] Department of Financial Engineering, Korea University, Seoul 02841, Republic of Korea

[2] Department of Mathematics, Korea University, Seoul 02841, Republic of Korea

[3] Department of Mathematics and Physics, Gangneung-Wonju National University, Gangneung 25457, State, Republic of Korea

[4] Program in Actuarial Science and Financial Engineering, Korea University, Seoul 02841, Republic of Korea