

Research Article

Reconstruction of the Time-Dependent Volatility Function Using the Black–Scholes Model

Yuzy Jin,¹ Jian Wang,² Sangkwon Kim,² Youngjin Heo,² Changwoo Yoo,³
Youngrock Kim ,⁴ Junseok Kim ,² and Darae Jeong ⁵

¹Department of Mathematics, Jilin Institute of Chemical Technology, Jilin 132022, China

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

³Department of Financial Engineering, Korea University, Seoul 02841, Republic of Korea

⁴Major in Mathematics Education, Hankuk University of Foreign Studies, Seoul 02450, Republic of Korea

⁵Department of Mathematics, Kangwon National University, Gangwon-do 24341, Republic of Korea

Correspondence should be addressed to Darae Jeong; tinayoyo@korea.ac.kr

Received 19 December 2017; Accepted 1 April 2018; Published 7 May 2018

Academic Editor: Francisco R. Villatoro

Copyright © 2018 Yuzy Jin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose a simple and robust numerical algorithm to estimate a time-dependent volatility function from a set of market observations, using the Black–Scholes (BS) model. We employ a fully implicit finite difference method to solve the BS equation numerically. To define the time-dependent volatility function, we define a cost function that is the sum of the squared errors between the market values and the theoretical values obtained by the BS model using the time-dependent volatility function. To minimize the cost function, we employ the steepest descent method. However, in general, volatility functions for minimizing the cost function are nonunique. To resolve this problem, we propose a predictor-corrector technique. As the first step, we construct the volatility function as a constant. Then, in the next step, our algorithm follows the prediction step and correction step at half-backward time level. The constructed volatility function is continuous and piecewise linear with respect to the time variable. We demonstrate the ability of the proposed algorithm to reconstruct time-dependent volatility functions using manufactured volatility functions. We also present some numerical results for real market data using the proposed volatility function reconstruction algorithm.

1. Introduction

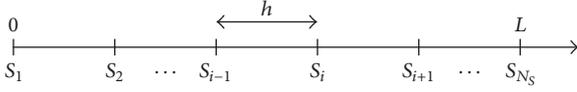
The accurate calibration of models using market option data is one of the most important problems in finance [1]. The reason is for accurate pricing and accordingly hedging strategy [2]. According to [3], the authors proposed the accurate computations for Greeks using the numerical solutions of the Black–Scholes partial differential equation. The well-known standard Black–Scholes (BS) model is not adequate for calibrating the market option data because it uses the constant volatility [4–6]. As an alternative to the BS equation with constant volatility [7], local volatility models were introduced to explain the volatility smiles or skews observed in the market. There has been much research carried out regarding the reconstruction of local volatility functions from market data [8, 9]. For example, in [10], radial basis functions were used to construct local volatility surfaces.

In [1], the authors proposed a regularized optimization formulation, using spline kernels to ensure both accuracy and stability in the local volatility function calibration. In [9], the authors mentioned the calibration of a local volatility surface for European options using a nonparametric approach by employing a second-order Tikhonov regularization.

The main goal of this study is to propose a new simple and robust numerical method for the construction of a time-dependent volatility function, using the BS partial differential equation with nonconstant volatility [11, 12]:

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

for $(S, t) \in \mathbb{R}^+ \times [0, T]$, where $u(S, t)$ is the option value of the underlying price S and time t , $\sigma(t)$ is the volatility function of

FIGURE 1: Uniform grid with a spatial step size h .

time t , and r is the riskless interest rate. The final condition is the payoff function $u(S, T) = \Lambda(S)$ at expiry T . Typically, the local volatility function is given as a function of an underlying asset and time, that is, $\sigma(S, t)$ [13, 14]. However, for simplicity and robustness of the solution algorithm, we assume that the local volatility function only depends on the time and is a piecewise linear function with respect to the time variable.

The outline of this paper is as follows. In Section 2, we describe our numerical algorithm for constructing the time-dependent volatility function. In Section 3, numerical experiments are presented. Finally, conclusions are drawn in Section 4.

2. Numerical Algorithm

In this section, we present a new numerical algorithm for constructing the time-varying volatility function.

2.1. Numerical Solution. Let S be the value of the underlying asset price and $\tau = T - t$ be the time to expiry, then (1) can be given as the following initial-value problem:

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

for $(S, \tau) \in \Omega \times (0, T]$ with an initial condition $u(S, 0) = \Lambda(S)$ for $S \in \Omega = (0, L)$, where the infinite domain is truncated to a finite computational domain [15]. Now, to solve (2) numerically, we apply a finite difference method (FDM). Let us denote the numerical approximation of the solution of (2) by $u_i^n \equiv u(S_i, \tau_n) = u(ih, n\Delta\tau)$, for $i = 1, 2, \dots, N_S$ and $n = 0, 1, \dots, N_\tau$. Here, $h = L/(N_S - 1)$ and $\Delta\tau = T/N_\tau$ are uniform spatial and temporal step sizes, respectively. N_S is the number of grid points and N_τ is the number of time steps. Figure 1 illustrates the uniform grid with a spatial step size h . Furthermore, the variable volatility σ^n is defined similarly as $\sigma^n \equiv \sigma(\tau_n)$.

By applying the fully implicit-in-time and space-centered difference scheme to (2), we obtain that

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

We can rewrite (3) as

$$\alpha_i u_{i-1}^{n+1} + \beta_i u_i^{n+1} + \gamma_i u_{i+1}^{n+1} = b_i, \quad \text{for } i = 2, \dots, N_S, \quad (4)$$

where $\alpha_i = rS_i/2h - (\sigma^{n+1} S_i)^2/2h^2$, $\beta_i = 1/\Delta\tau + (\sigma^{n+1} S_i)^2/h^2 + r$, $\gamma_i = -rS_i/2h - (\sigma^{n+1} S_i)^2/2h^2$, and $b_i = u_i^n/\Delta\tau$. For boundary

conditions, we use the zero Dirichlet condition at S_1 , that is, $u_1^n = 0$, and the linear condition at S_{N_S} , that is, $u_{N_S+1}^n = 2u_{N_S}^n - u_{N_S-1}^n$, for all n [16]. To solve the resulting discrete system (4), we apply the Thomas algorithm [17].

2.2. Algorithm of Volatility Construction. Next, we describe the proposed algorithm for constructing the time-dependent volatility using option prices. Suppose that we have a set of measurements $\{\omega_\beta^\alpha\}$, where ω_β^α is the market price of the options with the exercise time T_α for $\alpha = 1, \dots, M_t$ and the exercise price K_β for $\beta = 1, \dots, M_k$. Here, we assume that $T_1 \leq \dots \leq T_{M_t}$ and $K_1 \leq \dots \leq K_{M_k}$. Using the given data, we determine a volatility function $\sigma(t)$ in the least-squares sense. That is, we minimize the following mean-square error:

$$\Gamma_\alpha(\sigma) = \frac{1}{M_k} \sum_{\beta=1}^{M_k} \left[u_{K_\beta}(\sigma; S_0, T_\alpha) - \omega_\beta^\alpha \right]^2 \chi_\beta^\alpha \quad (5)$$

for $\alpha = 1, \dots, M_t$,

where $u_{K_\beta}(\sigma; S_0, T_\alpha)$ is the numerical solution at $S = S_0$ of 2 with the strike price K_β at time T_α . Here, χ_β^α is a characteristic function, which is equal to one if ω_β^α is available and otherwise zero. In this paper, we use the steepest descent method [18] to find the optimal value σ that minimizes the cost function $\Gamma_\alpha(\sigma)$. Now, we present the detailed process of the proposed algorithm. We use the notation $T_{\alpha+1/2} = (T_\alpha + T_{\alpha+1})/2$ for $\alpha = 0, 1, \dots$, where $T_0 = 0$.

Step 1 (find the constant volatility function on $[0, T_1]$). By assuming $\sigma(t) = y_1$ on $[0, T_1]$, we find that y_1 minimizes the following cost function:

$$\Gamma_1(\sigma) = \frac{1}{M_k} \sum_{\beta=1}^{M_k} \left[u_{K_\beta}(\sigma; S_0, T_1) - \omega_\beta^1 \right]^2 \chi_\beta^1. \quad (6)$$

After determining the volatility function, we denote $\sigma_1(t) := \sigma(t)$. Figure 2(a) illustrates the constant volatility function at this step.

Step 2 (find the linear volatility function on $[0, T_2]$). With y_1 obtained in **Step 1**, we define the linear volatility function $\sigma(t)$ on $[0, T_2]$ as

$$\sigma(t) = \frac{y_2 - y_1}{T_2 - 0.5T_1} (t - T_2) + y_2. \quad (7)$$

Here, (7) represents a linear function passing through the point $(T_1/2, y_1)$ and having the optimal value y_2 that minimizes the following cost function:

$$\Gamma_2(\sigma) = \frac{1}{M_k} \sum_{\beta=1}^{M_k} \left[u_{K_\beta}(\sigma; S_0, T_2) - \omega_\beta^2 \right]^2 \chi_\beta^2. \quad (8)$$

Note that, by setting (7), we only estimate a single parameter y_2 using the given value for y_1 . After determining the volatility function, we denote $\sigma_2(t) := \sigma(t)$. Figure 2(b) illustrates the linear volatility function $\sigma(t)$ on $[0, T_2]$.

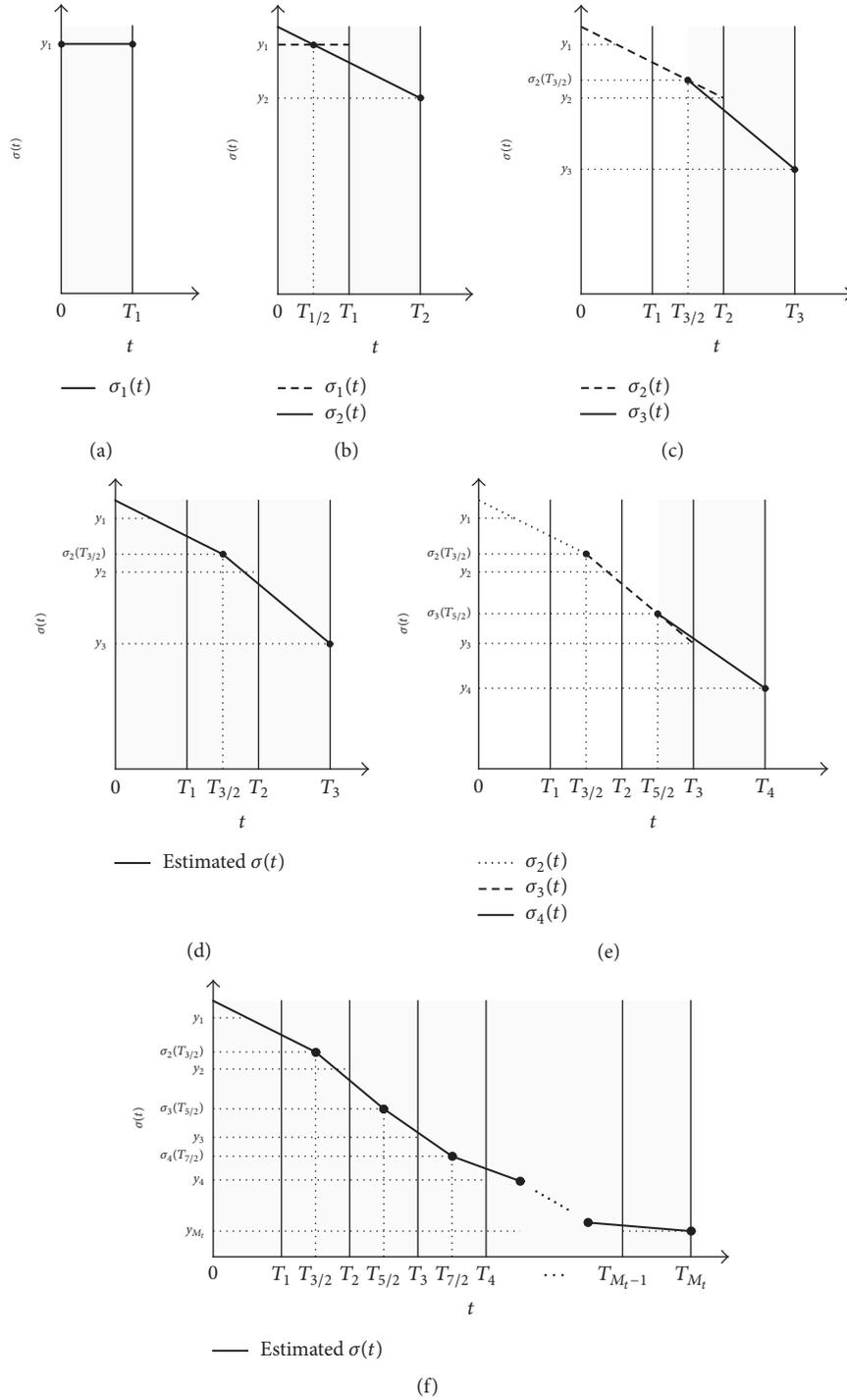


FIGURE 2: Schematic of procedure for time-dependent volatility function.

Step 3 (find the piecewise linear volatility function on $[0, T_3]$). First, we assume that the linear volatility function $\sigma_3(t)$ on $[T_{3/2}, T_3]$ is given by

$$\sigma_3(t) = \frac{y_3 - \sigma_2(T_{3/2})}{T_3 - T_{3/2}}(t - T_3) + y_3, \quad (9)$$

where the linear equation passes through the given point $(T_{3/2}, \sigma_2(T_{3/2}))$ and unknown point (T_3, y_3) as shown in

Figure 2(c). From the linear function σ_3 , we define the following piecewise linear volatility function on $[0, T_3]$:

$$\sigma(t) = \begin{cases} \sigma_2(t), & \text{if } t \in [0, T_{3/2}], \\ \sigma_3(t), & \text{if } t \in [T_{3/2}, T_3]. \end{cases} \quad (10)$$

Then, we estimate the optimal value of y_3 that minimizes the cost function as

$$\Gamma_3(\sigma) = \frac{1}{M_k} \sum_{\beta=1}^{M_k} \left[u_{K_\beta}(\sigma; S_0, T_3) - \omega_\beta^3 \right]^2 \chi_\beta^3. \quad (11)$$

Figure 2(d) presents a schematic of the piecewise linear volatility function on $[0, T_3]$.

Step 4 (find the linear volatility function on $[0, T_{M_t}]$). The following process is repeated from $\alpha = 4$ to $\alpha = M_t$. By using $y_{\alpha-1}$ as obtained in previous step, we set the linear volatility function $\sigma_\alpha(t)$ on $[T_{\alpha-3/2}, T_\alpha]$ as

$$\sigma_\alpha(t) = \frac{y_\alpha - \sigma_{\alpha-1}(T_{\alpha-3/2})}{T_\alpha - T_{\alpha-3/2}} (t - T_{\alpha-3/2}) + y_\alpha. \quad (12)$$

In general, by using the volatility function

$$\sigma(t) = \begin{cases} \sigma_2(t) & \text{if } t \in [0, T_{3/2}], \\ \sigma_j(t) & \text{if } t \in [T_{j-3/2}, T_{j-1/2}] \text{ for } 2 < j < \alpha, \\ \sigma_\alpha(t) & \text{if } t \in [T_{\alpha-3/2}, T_\alpha], \end{cases} \quad (13)$$

we determine the optimal parameter y_α that minimizes the cost function as

$$\Gamma_\alpha(\sigma) = \frac{1}{M_k} \sum_{\beta=1}^{M_k} \left[u_{K_\beta}(\sigma; S_0, T_\alpha) - \omega_\beta^\alpha \right]^2 \chi_\beta^\alpha. \quad (14)$$

Note that in the proposed algorithm we do not compute the implied volatility and do not use Dupire's formula [12, 19–21]. Instead, we directly compute the time-dependent volatility function from option prices and the BS equation. Our reconstructed volatility function is piecewise linear. It is very robust to compute the volatility because we only need to find a single parameter at each steepest descent algorithm.

3. Numerical Experiments

In this section, we demonstrate the performance of the proposed time-dependent volatility construction algorithm by numerical experiments with manufactured volatility functions and option data from a real market. All computations were performed on a 2.7 GHz Intel PC with 8 GB of RAM loaded with MATLAB 2016a [22].

3.1. Convergence Test. As the first numerical test, we perform a convergence test for the numerical scheme (4) with respect to h and $\Delta\tau$. Let us consider a payoff function $\Lambda(S) = \max(S - 100, 0)$ with $r = 0.015$, $S_0 = 100$, $T = 1$, and $L = 400$. We set a complicated volatility function, $\sigma(t) = 0.1 \cos(4\pi t) - 0.1t + 0.2$ (see Figure 3), to check the convergence of the employed numerical scheme.

Table 1 shows the convergence of European call option prices at (S_0, T) as we refine h and $\Delta\tau$. From this point on, we will use $h = 1$ and $\Delta\tau = 1/360$, because these values are sufficiently accurate for the present work, and we will use $r = 0.015$ and $L = 400$ unless otherwise specified.

TABLE 1: Convergence test for European call option values.

Case	$h = 4$	$h = 2$	$h = 1$	$h = 0.5$
$\Delta\tau = 1/90$	7.397038	7.432546	7.441332	7.443523
$\Delta\tau = 1/180$	7.389061	7.424523	7.433299	7.435488
$\Delta\tau = 1/360$	7.385044	7.420485	7.429255	7.431443
$\Delta\tau = 1/720$	7.383028	7.418458	7.427226	7.429413

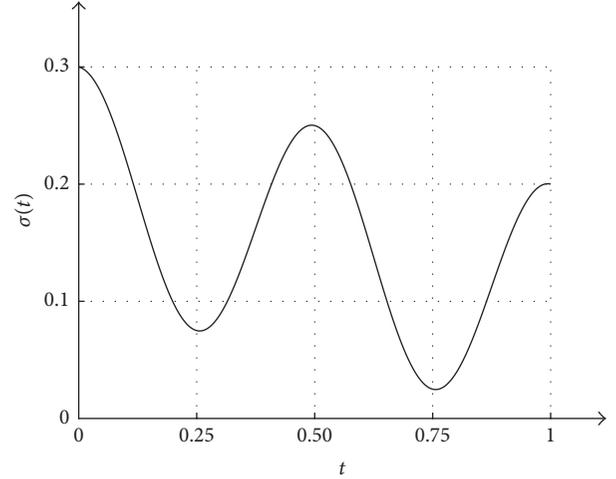


FIGURE 3: Volatility function, $\sigma(t) = 0.1 \cos(4\pi t) - 0.1t + 0.2$.

3.2. Nonuniqueness of the Volatility Function. We consider a numerical test to demonstrate the nonuniqueness of a linear volatility function. Let $\sigma(t) = -0.2t + 0.3$ be a given volatility function on a time interval $[0, 1]$. Using this given volatility function, we generate European call prices by solving (4). The other parameters used are $S_0 = 100$ and $K_\beta = 92 + 2\beta$ for $\beta = 1, \dots, 8$. The payoff functions are set as $\Lambda_\beta(S) = \max(S - K_\beta, 0)$ for $\beta = 1, \dots, 8$. Table 2 presents the European call option prices that are generated using the given conditions.

Now, we will determine the following linear volatility function which minimizes the cost function:

$$\sigma(t) = (\sigma_1 - \sigma_0)t + \sigma_0, \quad (15)$$

where $\sigma_0 = \sigma(0)$ and $\sigma_1 = \sigma(1)$ are constants to be computed. Figure 4(a) shows the cost function $\Gamma(\sigma) = \sum_{\beta=1}^8 [u_{K_\beta}(\sigma; 100, 1) - \omega_\beta]^2 / 8$ on a parameter domain, $0.1 \leq \sigma_0, \sigma_1 \leq 0.3$. Figure 4(b) shows contour lines of the cost function $\Gamma(\sigma)$. As shown in Figure 4(b), we obtain the different points (star markers) having the local minimum of the cost function $\Gamma(\sigma)$ by applying the steepest descent method with different initial guesses (from A to G). In Figure 4(c), we represent these estimated linear volatility functions (15), satisfying $\Gamma(\sigma) \approx 0$. This means that the optimal parameter values for (15) are nonunique. Therefore, in the first step of the construction of the volatility function, it is reasonable to assume a constant volatility function.

3.3. Volatility Construction Using Manufactured Data. Next, we consider the following nonlinear manufactured volatility function:

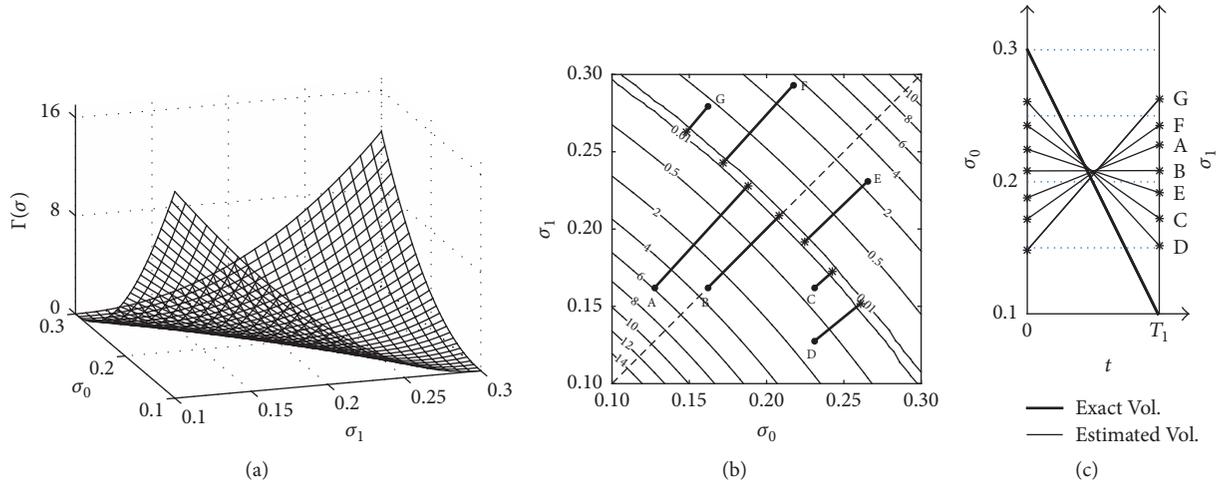


FIGURE 4: (a) Cost function $\Gamma(\sigma) = \sum_{\beta=1}^8 [u_{K_\beta}(\sigma; 100, 1) - \omega_\beta]^2 / 8$ on a parameter domain, $0.1 \leq \sigma_0, \sigma_1 \leq 0.3$. (b) Contour lines of the cost function and behavior of eight different points. (c) Estimated volatility functions for corresponding points.

TABLE 2: Generated European call option prices for different strike prices K_β .

β	1	2	3	4	5	6	7	8
K_β	94	96	98	100	102	104	106	108
ω_β	12.22	11.07	10.00	9.00	8.08	7.23	6.45	5.74

TABLE 3: European call option prices generated by the volatility function $\sigma(t) = 0.3/3^t$.

K_β	94	96	98	100	102	104	106	108
$T_1 = 90/360$	8.87	7.60	6.45	5.42	4.51	3.72	3.05	2.47
$T_2 = 180/360$	10.27	9.06	7.95	6.94	6.03	5.20	4.47	3.82
$T_3 = 270/360$	11.08	9.89	8.79	7.78	6.86	6.02	5.26	4.58
$T_4 = 360/360$	11.61	10.43	9.33	8.32	7.39	6.54	5.76	5.06

$$\sigma(t) = \frac{0.3}{3^t}. \tag{16}$$

First, we obtain reference values for the call option that is based on the given volatility function (16) by solving (4) with $T_\alpha = 90\alpha/360$ for $\alpha = 1, 2, 3, 4$ and $K_\beta = 92 + 2\beta$ for $\beta = 1, 2, \dots, 8$. The European call option prices generated by the volatility function (16) are represented in Table 3.

In Figure 5, we see the procedure for constructing the volatility function up until $t = T_4$. Here, we denote the temporal volatility function $\sigma(t)$ at each time level $t = T_i$

by solid lines and star markers. Also, we have inserted small figures to illustrate the cost function $\Gamma(\sigma)$ against the volatility and highlighted the optimal point that minimizes $\Gamma(\sigma)$ at $t = T_i$ for $i = 1, 2, 3, 4$. From this figure, we can see that our proposed method satisfactorily recovers the exact volatility function.

3.4. Basic Mechanism of the Proposed Algorithm. In this section, we introduce the basic mechanism of the proposed algorithm. For comparison, we consider a simple piecewise linear (SPL) volatility function:

$$\sigma_{\text{SPL}}(t) = \begin{cases} \sigma_1(t) = y_1 & \text{if } t \in [0, T_1], \\ \sigma_\alpha(t) = \frac{y_\alpha - y_{\alpha-1}}{T_\alpha - T_{\alpha-1}}(t - T_{\alpha-1}) + y_{\alpha-1} & \text{if } t \in [T_{\alpha-1}, T_\alpha] \text{ for } \alpha = 2, \dots, M_t. \end{cases} \tag{17}$$

If we apply the SPL volatility function (17) to approximate the two exact volatility functions $\sigma(t) = 0.3/3^t$ and $\sigma(t) = 0.1 \cos(4\pi t) - 0.1t + 0.2$, then we obtain the results (dashed lines) illustrated in Figure 6. In Figure 6(a), the volatility is constant in the first interval, and in the second interval the value of y_2 should be smaller than the exact value $\sigma(T_2)$ to minimize the cost function. Likewise, y_3 is larger than $\sigma(T_3)$ and y_4 is again smaller than $\sigma(T_4)$. Figure 6(b) shows similar behavior; that is, an oscillatory solution is

generated. However, the proposed algorithm for constructing the volatility function is essentially a predictor-corrector technique. It dampens spurious oscillations and generates satisfactory results. As shown in Figure 6, the results (thin solid lines) using the proposed algorithm are accurate and robust.

3.5. Calculate for the Whole Time at Once. Until now, we have found an optimized volatility for each section of time. But

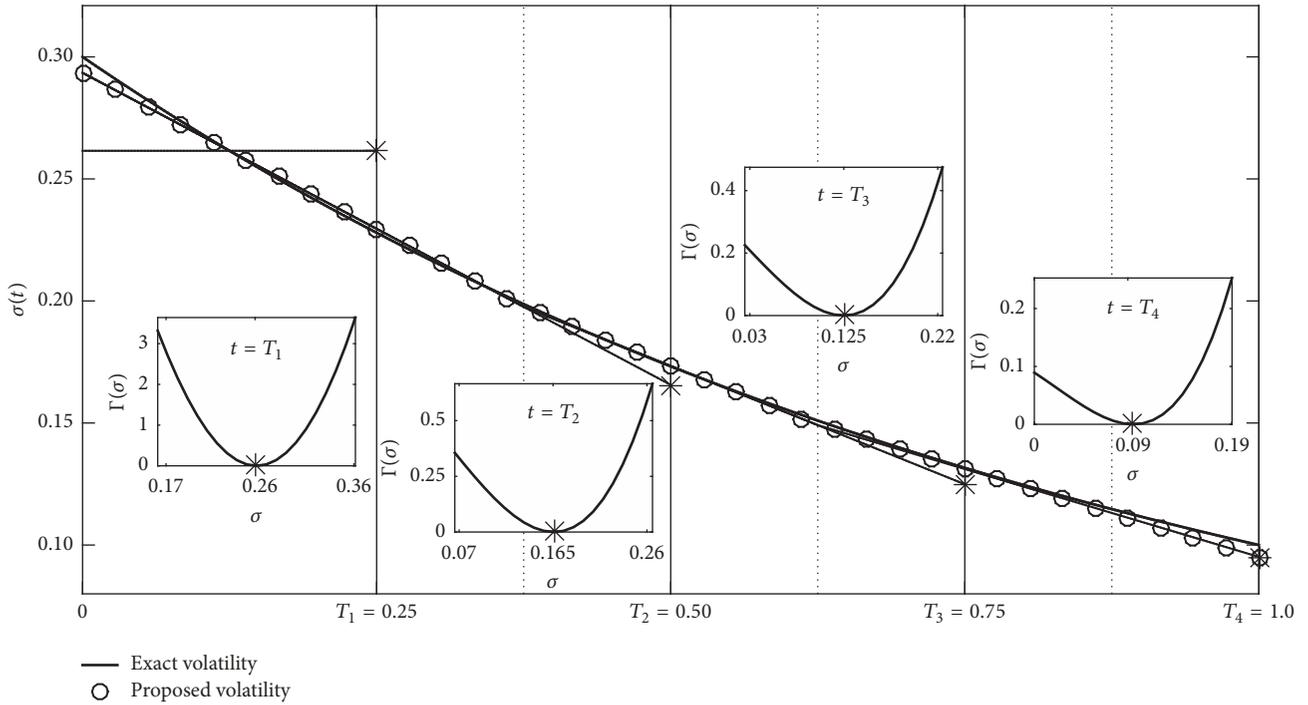


FIGURE 5: Estimated time-dependent volatility by our algorithm and the exact volatility function (16). Here, each small figure shows the cost function $\Gamma(\sigma)$ as a function of σ and highlights the optimal value that minimizes $\Gamma(\sigma)$ at time level $t = T_\alpha$ for $\alpha = 1, 2, 3, 4$.

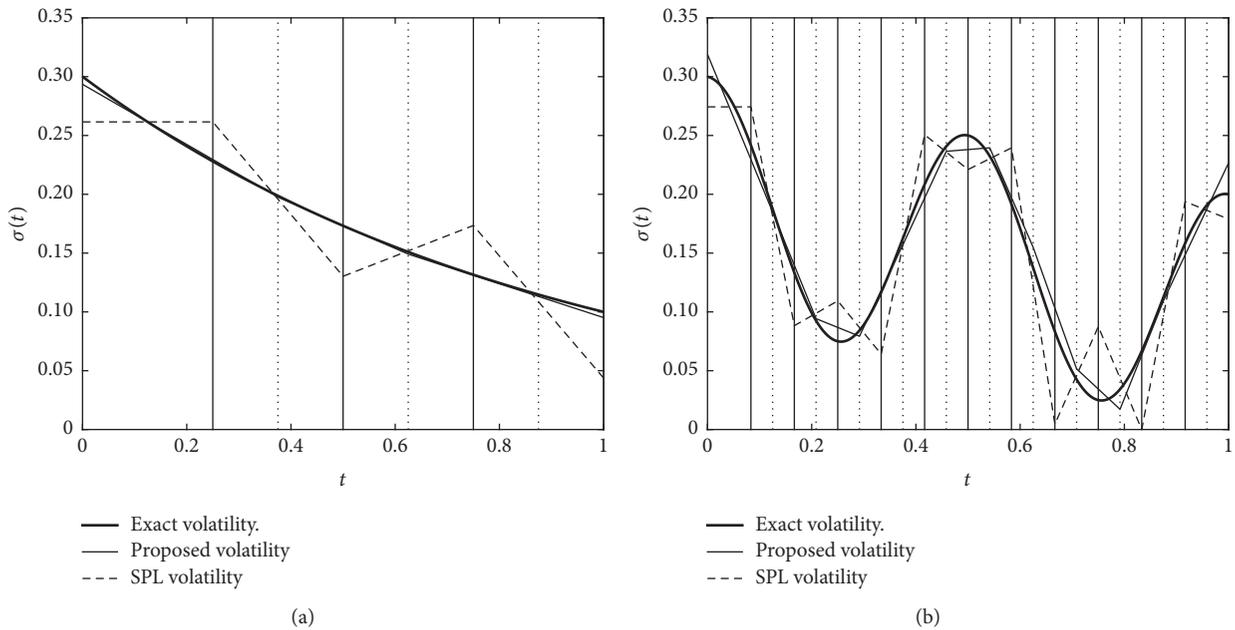


FIGURE 6: Construction of volatility by the SPL and proposed algorithms when the exact volatility is (a) $\sigma(t) = 0.3/3^t$ and (b) $\sigma(t) = 0.1 \cos(4\pi t) - 0.1t + 0.2$.

from now on, we will not calculate the time for each section but the volatility for all of the time. The comparison between the two results is shown in Figure 7. Table 4 shows the value of $\Gamma(\sigma)$.

3.6. Volatility Construction from KOSPI 200 Data. Now, we construct the volatility function by the proposed algorithm with the KOSPI 200 index call option data [23, 24] on 29 July 2016. Table 5 shows real market data for the KOSPI 200 index

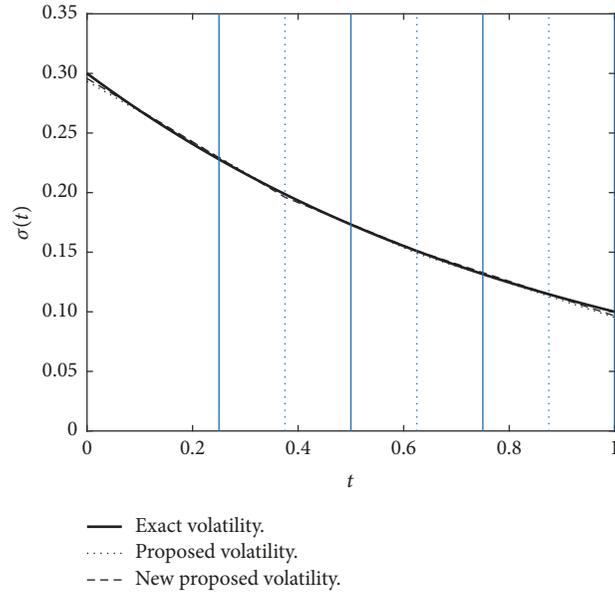


FIGURE 7: Construction of volatility by proposed algorithm and new proposed algorithm when the exact volatility is $\sigma(t) = 0.3/3^t$.

TABLE 4: The value of $\Gamma(\sigma)$.

	Proposed volatility	New proposed volatility
$\Gamma(\sigma)$	$1.4796e - 04$	$6.1036e - 06$

TABLE 5: KOSPI 200 index call option price on 29 July 2016 with respect to the strike and maturity.

K_β	245.0	247.5	250.0	252.5	255.0	257.5	260.0	262.5
$T_1 = 13\Delta\tau$	7.34	5.29	3.39	1.93	0.92	0.37	0.11	0.04
$T_2 = 41\Delta\tau$	8.85	6.90	5.28	3.82	2.56	1.70	1.05	0.62
$T_3 = 76\Delta\tau$	9.34	8.41	6.95	4.72	4.24	3.21	2.36	1.74

call option price with respect to the strike and maturity. Here, the strike indices are $K_\beta = 245 + 2.5(\beta - 1)$ for $\beta = 1, 2, \dots, 8$ and the maturity times are $\Delta\tau = 1/365$, $T_1 = 13\Delta\tau$, $T_2 = 41\Delta\tau$, and $T_3 = 76\Delta\tau$. At this time, the current value of the KOSPI 200 index was $S_0 = 251.48$ and interest rate was $r = 0.0136$.

By the proposed algorithm, we construct the time-dependent volatility function $\sigma(t)$ from the KOSPI 200 market data as shown in Figure 8(d). Figures 8(a)–8(c) represent the numerical option values given by the proposed volatility function and the real market data for the KOSPI 200 index at $t = T_1, T_2$, and T_3 .

4. Conclusions

In this paper, we considered a simple and robust numerical algorithm for the construction of a time-dependent volatility function from a finite set of market observations by the

Black–Scholes model. In order to numerically solve the BS equation, we applied a fully implicit finite difference method. By assuming a continuous and piecewise linear function with respect to time, we constructed a volatility function at each time level. The construction algorithm for the volatility function is based on the minimization of a cost function, defined by the sum of the squared errors between market values and theoretical values obtained from the BS model using the time-dependent volatility function. Here, we used a steepest descent method to minimize the cost function. However, in general, volatility functions for minimizing the cost function are nonunique. To resolve this problem, we proposed a predictor-corrector technique. As the first step, we construct the volatility function as a constant. Then, in the next step, our algorithm follows the prediction step and correction step at half-backward time level. Using several examples, we demonstrated the accuracy and robustness of our proposed algorithm.

Disclosure

An earlier version of the manuscript was presented as a poster at SIAM 2017, Spring Conference, Seoul National University, Seoul, Korea, June 23–24, 2017.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The first author (Yuzi Jin) was supported by the University-Level Major Social Science Project of Jilin Institute of Chemical Technology, no. 20, 2016. The author (Youngrock Kim)

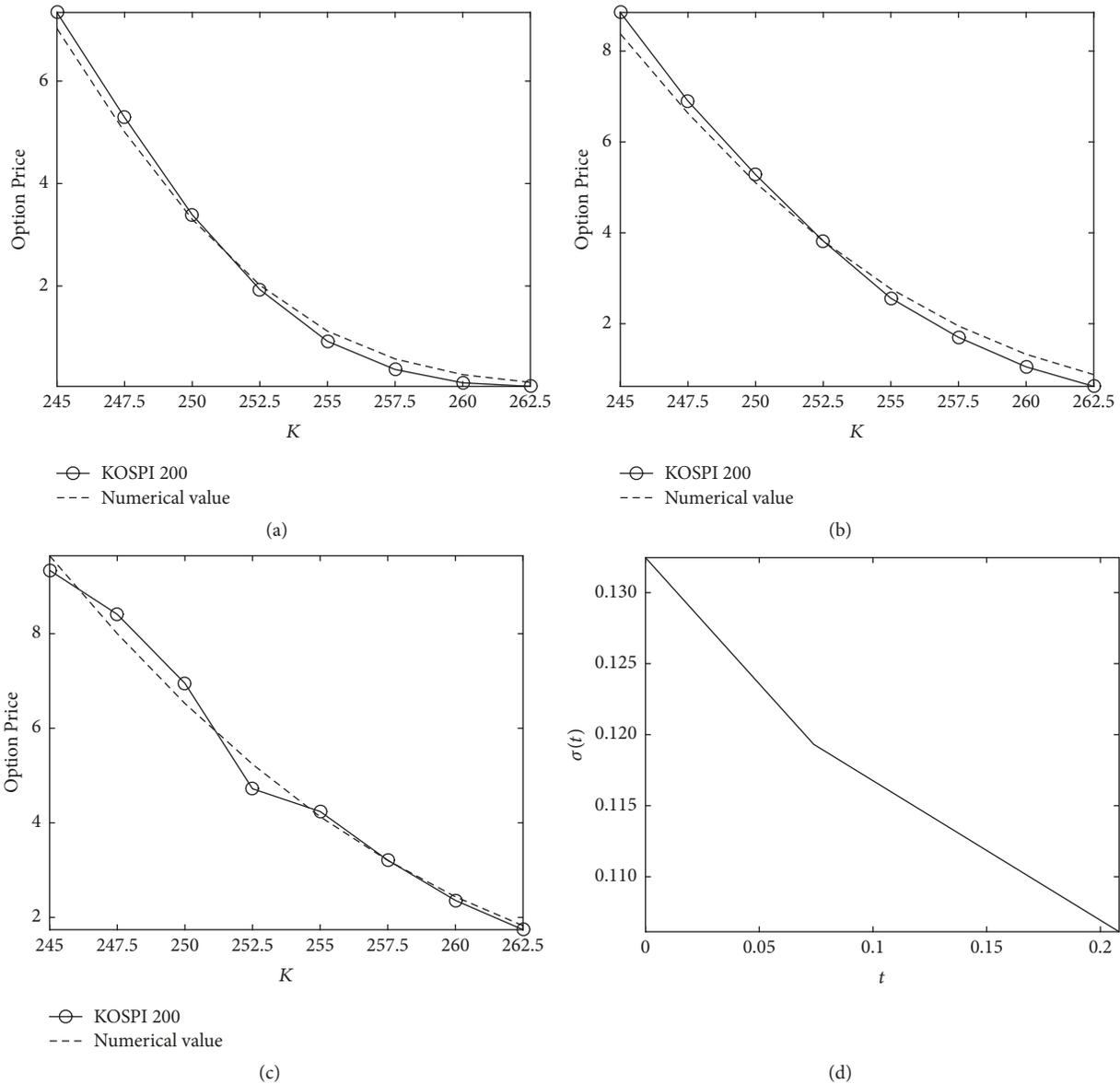


FIGURE 8: Comparison of KOSPI 200 index call option price (circle-marked solid line) with numerical option prices (dashed line) at (a) $T = 13/365$, (b) $41/365$, and (c) $76/365$. (d) Constructed time-dependent volatility function.

was supported by the Basic Science Research Program of the NRF (Korea) under Grant no. 2015RID1A1A01059643/2.

References

- [1] T. F. Coleman, Y. Li, and C. Wang, "Stable local volatility function calibration using spline kernel," *Computational optimization and applications*, vol. 55, no. 3, pp. 675–702, 2013.
- [2] C. Liang and S. Li, "Pricing and hedging European-style options in Lévy-based stochastic volatility models considering the leverage effect," *Journal of Mathematical Analysis and Applications*, vol. 438, no. 2, pp. 1010–1029, 2016.
- [3] D. Jeong, M. Yoo, and J. Kim, "Accurate and efficient computations of the Greeks for options near expiry using the Black-Scholes equations," *Discrete Dynamics in Nature and Society*, vol. 2016, Article ID 1586786, 12 pages, 2016.
- [4] M. Rubinstein, "Implied binomial trees," *The Journal of Finance*, vol. 49, no. 3, pp. 771–818, 1994.
- [5] C. Fan, K. Xiang, and P. Chen, "Efficient option pricing in crisis based on dynamic elasticity of variance model," *Discrete Dynamics in Nature and Society*, vol. 2016, Article ID 7496539, 9 pages, 2016.
- [6] P. Li and J. Yang, "Pricing collar options with stochastic volatility," *Discrete Dynamics in Nature and Society*, vol. 2017, Article ID 9673630, 7 pages, 2017.
- [7] F. Black and M. Scholes, "The pricing of options and corporate liabilities," *Journal of Political Economy*, vol. 81, no. 3, pp. 637–654, 1973.
- [8] Z.-C. Deng, Y. C. Hon, and V. Isakov, "Recovery of time-dependent volatility in option pricing model," *Inverse Problems*, vol. 32, no. 11, Article ID 115010, 30 pages, 2016.

- [9] J. Geng, I. M. Navon, and X. Chen, “Non-parametric calibration of the local volatility surface for European options using a second-order Tikhonov regularization,” *Quantitative Finance*, vol. 14, no. 1, pp. 73–85, 2014.
- [10] J. Glover and M. M. Ali, “Using radial basis functions to construct local volatility surfaces,” *Applied Mathematics and Computation*, vol. 217, no. 9, pp. 4834–4839, 2011.
- [11] E. Derman and I. Kani, “Riding on a smile,” *Risk Magazine*, vol. 7, no. 2, pp. 32–39, 1994.
- [12] B. Dupire, “Pricing with a smile,” *Risk Magazine*, vol. 7, no. 1, pp. 18–20, 1994.
- [13] S. Crepey, “Calibration of the local volatility in a generalized Black-Scholes model using Tikhonov regularization,” *SIAM Journal on Mathematical Analysis*, vol. 34, no. 5, pp. 1183–1206, 2003.
- [14] Z. Cen and A. Le, “A robust and accurate finite difference method for a generalized Black-Scholes equation,” *Journal of Computational and Applied Mathematics*, vol. 235, no. 13, pp. 3728–3733, 2011.
- [15] D. Tavella and C. Randall, *Pricing Financial Instruments: The Finite Difference Method*, John Wiley and Sons, New York, NY, USA, 2000.
- [16] H. Windcliff, P. Forsyth, and K. Vetzal, “Analysis of the stability of the linear boundary condition for the Black-Scholes equation,” *The Journal of Computational Finance*, vol. 8, no. 1, pp. 65–92, 2004.
- [17] L. Thomas, *Elliptic Problems in Linear Differential Equations over a Network*, Watson Scientific Computing Laboratory, Columbia University, New York, NY, USA, 1949.
- [18] R. L. Burden and J. D. Faires, *Numerical Analysis*, Thomson Brooks Cole, Boston, Mass, USA, 2001.
- [19] P. Carr, H. Geman, D. B. Madan, and M. Yor, “From local volatility to local Lévy models,” *Quantitative Finance*, vol. 4, no. 5, pp. 581–588, 2004.
- [20] G. Deelstra and G. Rayée, “Local volatility pricing models for long-dated FX derivatives,” *Applied Mathematical Finance*, vol. 20, no. 4, pp. 380–402, 2013.
- [21] P. K. Friz, S. Gerhold, and M. Yor, “How to make Dupire’s local volatility work with jumps,” *Quantitative Finance*, vol. 14, no. 8, pp. 1327–1331, 2014.
- [22] The MathWorks Inc, *MATLAB*, The MathWorks Inc, Natick, Mass, USA, 1994, <http://www.mathworks.com/>.
- [23] H.-J. Ahn, J. Kang, and D. Ryu, “Informed trading in the index option market: The case of KOSPI 200 options,” *Journal of Futures Markets*, vol. 28, no. 12, pp. 1118–1146, 2008.
- [24] D. Ryu, “Implied volatility index of KOSPI200: Information contents and properties,” *Emerging Markets Finance and Trade*, vol. 48, no. 2, pp. 24–39, 2012.



Hindawi

Submit your manuscripts at
www.hindawi.com

