Journal of Industrial and Management Optimization
Vol. 20, No. 12, December 2024, pp. 3792-3815
doi:10.3934/jimo.2024071

SOLVING AMERICAN OPTION OPTIMAL CONTROL PROBLEMS IN FINANCIAL MARKETS USING A NOVEL NEURAL NETWORK

Jiao Teng[1], Kit Yan Chan[2] and Ka Fai Cedric Yiu[*1]

[1]Department of Applied Mathematics,
The Hong Kong Polytechnic University, Hunghom, Kowloon, Hong Kong, China

[2]School of Electrical Engineering, Computing and Mathematical Sciences,
Curtin University, Perth, WA6102, Australia

(Communicated by Bin Li)

Abstract. In this paper, we introduce a novel neural network (NN) for solving optimal control problems associated with American options in financial markets. American options provide holders with the flexibility to exercise the option before expiration, thereby affecting potential profitability. This paper focuses on determining the optimal exercise strategy and option price to maximize the payoff by solving a class of American option optimal control problems. We reformulate the optimal control problem into a *linear complementarity problem*(LCP). Subsequently, we employ the penalty approach and smoothing method to convert the LCP into a bi-nonlinear system with a set of partial differential equations (PDEs). By solving the reformulated PDE equations with the proposed method, we obtain numerical solutions that yield the optimal exercise strategy and option price. Numerical examples of American call and put options demonstrate the efficiency and usefulness of the proposed methods.

1. Introduction.

In recent years, Artificial Intelligence (AI) has grown rapidly. Neural networks, as a key part of machine learning algorithms, are widely used in numerous fields, from image analysis [21], scientific computing [15], natural language processing [2], noise reduction [36], and finance [14]. At the same time, research has been conducted to use neural networks for solving optimal control problems, such as a tracking control algorithm based on fuzzy reinforcement learning [30] or trajectory tracking control using neural networks [9].

Optimizing the timing for buying or selling an asset is crucial for trading and risk management in financial markets. Therefore, the study of the optimal control problem for American options is essential, as it can assist in determining the optimal exercise strategy and option value, providing vital insights and decision support for trading and risk management in financial markets. American options

2020 *Mathematics Subject Classification.* Primary: 49M37, 91G60, 68T07; Secondary: 91G80.
Key words and phrases. Optimal control problem, linear complementarity problem, neural networks, American option, partial differential equation solver.

*Corresponding author: Ka Fai Cedric Yiu.

allow for strikes at any time prior to the expiry date, which introduces a complex optimal control problem, i.e., the optimal stopping time problem. Solving the optimal control problems attempts to determine how the dynamical systems are controlled with respect to a specific loss function [20]. Pontryagin's maximum principle depends on the solution of the two-point margin problem, which is not an easy problem to solve [23]. Dynamic programming methods face the curse of dimensionality and large time complexity [3]. Hence, numerical solution techniques incorporating modern computing capability is essential to solve real world problems with high computational complexity. In finance, buying or selling time of certain assets can be optimized by solving an optimal control problem. For example, an American option can be exercised at any time before the expiration date. In order to determine its value at the moment of optimal exercise time, we need to tackle the optimal stopping time problem to give the optimal exercise strategy and the expected discounted return of the option.

The American option problem can be solved by the Monte Carlo method [7], binomial method [10], and PDE method [37]. However, the computational cost of the Monte Carlo method increases rapidly when the number of underlying assets increases. The convergence of the binomial method is rather slow; higher accuracy requires sufficiently long spatial truncation, which increases the storage space and computational effort. The PDE method is a widely used and effective method. Particularly, the PDE method is very effective for solving the corresponding linear complementarity problem. In recent years, neural networks have been developed to determine American options [25]. Gaspar et al. [13] demonstrated that the NN method is better than the least-squares Monte Carlo (LSM) method. Anderson et al. [1] employed deep neural networks (DNN) to solve the American option pricing problem, and numerical results show that the DNN method is better than the finite difference PDE method and binomial method. However, these methods do not provide the essential exercise strategy and do not take full advantage of the option information. In our work, we consider formulating the American option as an optimal control problem. By solving the converted LCP, the optimal exercise strategy and the American option value can be solved simultaneously.

The solution of the LCP holds great practical importance in the field of computational mathematics. Despite existing analytical methods, numerous methods have been developed to solve variational inequality problems or LCP over the past decades. These methods include the interior point method [8], quadratic programming method [19], Newton method [35], path searching method [12], and other modified methods [28]. The common limitation of these methods is that complex transformations based on the expertise of specialists are required before performing the computations, although finite difference methods (FDMs) and finite element methods (FEMs) can be used to approximate solutions for boundary-value problems in PDEs [17]. These methods also encounter certain challenges. First, when solving high-dimensional problems, discretizing the high-dimensional space is unfeasible, and the "dimensionality disaster" is caused. Additionally, the computation at each discrete node depends solely on information from its neighboring nodes, and the localization limitation is caused. Moreover, denser grids are required to achieve high computational accuracy; thereby, more computation and storage are required.

Data-driven methods such as machine learning usually requires large amounts of data to develop the prediction models. However, the cost of collecting data is too expensive by performing experiments. Instrumentation measures are generally not accurate enough to obtain precise data with sufficient quality. In 2019, M. Raissi et al. [24] proposed physics-informed neural networks (PINNs) method which combines data and PDEs to solve mathematical physics problems. Neural networks and physical modeling are used to solve PDEs problems. Currently, PINNs are applied to solve many physics problems, such as fluid dynamics, heat conduction, quantum physics, and solid mechanics. These physics problems are solved using known physical laws and observational data, and physical equations are embedded into neural networks, enabling the neural networks to directly learn and predict the behavior of physical systems. Otherwise, PINNs are able to solve PDEs which are too demanding to be solved by numerical simulations, as well as inverse problems and constrained optimization [4]. Motivated by this approach, we propose a novel neural network to solve the American option optimal control problem. In the proposed method, the structural properties of the problem are integrated to train the neural network.

The proposed method overcomes the limitations of the commonly used methods. First, same as the meshless method, the proposed method eliminates the need for meshing the solution area in order to avoid dimensionality disasters which are often encountered in high-dimensional problems. Notably, our approach is able to take into account the underlying structural properties of the problem rather than relying solely on data. Hence, a more accurate and reliable solution is guaranteed to be obtained. Additionally, the commonly used methods usually require experts to transform the problems based on experience. In contrast, the proposed method does not require too much specialized knowledge and has a lower threshold for use. Moreover, in addition to deriving the option value, the proposed method provides crucial insights into the optimal exercise time for the option. The proposed method enables investors to make well-informed decisions and optimize their returns.

This paper attempts to determine the numerical solutions for the optimal control problem of American options. The optimal control problem is first reformulated as a *linear complementarity problem (LCP)*. To address the LCP, we employ the penalty function-based method and smoothing technique to convert LCP as the bi-nonlinear PDEs. Subsequently, we apply the novel NN method to determine the numerical solutions of the bi-nonlinear PDE equation. By solving the optimal control problem, we can determine the optimal exercise strategy and option value of the American option.

The main contribution of this article is summarized as follows:

1. A novel NN method is developed for solving the LCP. This method can be applied to determine the numerical solution of the American options pricing problem.
2. By solving the optimal control problem associated with American options, we can obtain the optimal exercise strategy with the option price. This strategy has significant and practical implications since it allows holders to intelligently manage their assets and maximize their returns based on this strategy.
3. In contrast to traditional methods, which require experts to devise problem-specific transformations, the proposed NN-based method provides a simpler and adaptable approach in order to solve high-dimensional problems and QVI (quasi-variational inequality) problems.

The rest of this paper is organized as follows. In Section 2, we present the general optimal control problem of the American options and prove that the problem can be reformulated as a LCP. In Section 3, we apply the penalty function approach and soothing technique to the LCP and we reformulate it as a bi-nolinear system of PDE equations. Finally, we use the proposed NN algorithm to solve the reformulated problem. Section 4 uses three examples to demonstrate the effectiveness and validity of the proposed NN method.

2. **Optimal control problem of American options.** Financial derivatives, including options, are widely traded in financial markets. Among them, options are the most fundamental and widely used. Option pricing is essential in finance, as it addresses various market trading and risk management aspects. Given an asset price of $S(t)$ at time t, the key questions from a financial perspective are: How to determine the value V of the option? How to determine the optimal timing for exercising the option in order to maximize returns?

European or American options are categorized by whether they can be performed in advance or not. European options can only be exercised on the expiration date, while American options can be exercised at any time before expiration. In the absence of dividends, early exercise of American call options is typically not the optimal choice. Hence, the values for both American call and European options are equivalent. Conversely, for American put options, an optimal stopping time may exist before the expiration.

American options give the holder the right to exercise the options at a specified future time. Therefore, determining the exercise timing for American put options is critical [5]. In fact, the American option pricing problem is a free boundary problem. In financial engineering, it is also known as the optimal exercise boundary. Mathematically, it can be formulated by the Black-Scholes equation and is solved by identifying the best exercise curve represented by the free boundary. In the context of optimal control, the pricing of American put options can be formulated as an optimal control problem, where the option value is the optimal value and the optimal exercise boundary is the optimal solution. Prior to solving this optimal control problem, this section introduces the relevant background information and key concepts associated with American options.

2.1. **American options.** Assuming that the market is in a complete state with no arbitrage opportunities. S_t indicates the ex-dividend stock price at time t, behaves based on the following geometric Brownian motion:

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

where constants $r > 0$ and $\sigma > 0$ represent the growth rates without dividend and volatility of the stock, respectively. W_t is a standard Brownian motion on a probability space $(\Omega, \mathcal{F}, \mathbb{Q})$, Ω is a nonempty set, $\{\mathcal{F}_t\}_{t\geq 0}$ denotes the filtration generated by the stock price process, and $\mathbb{Q}$ denotes the Risk Neutral Measure.

The concept of the stopping time and exercise time is described prior introducing the problem formulation.

Definition 2.1. A random variable $\tau : \Omega \to [0, \infty]$ is called a stopping time for the filtration $\mathcal{F}_t$ if

$$\{\tau \leq t\} \triangleq \{\omega \in \Omega : \tau(\omega) \leq t\} \in \mathcal{F}_t, \text{ for any } t \geq 0.$$

When $t < T$, $\mathcal{F}_t$ is a set of all events determined in the first t trading periods where the price information is between $[0, t]$.

American options provide option holders with the freedom to exercise the option at any time before the expiration date. The holders are allowed to select the optimal exercise timing to maximize their benefits. Another approach to frame the problem of American options is to reformulate the problem as an *optimal control problem* in Problem 2.2. In this approach, the main objective is to solve the optimal control problem, in order to obtain two crucial components, namely the optimal value (i.e., the American option value) and the optimal solution (i.e., the optimal exercise boundary).

Problem 2.2. ***Optimal control problem of American option***

$$\begin{aligned} \max_{\tau \in \mathcal{T}_t} \quad & V(t, S_t) = \mathrm{E}\left[e^{-r(\tau - t)}(K - S(\tau))^+ | \mathcal{F}_t\right], \\ & s.t. \quad dS_t = rS_t dt + \sigma S_t dW_t, \end{aligned}$$

where T is the expiry time, and $\mathcal{F}_t \in [0, T]$ is a set of all stopping times. K is the strike price, indicating whether the option holder can buy or sell the stock when exercising the option contract. E[.|.] *is denoted as the conditional mathematical expectations with the risk-neutral measure $\mathbb{Q}$.*

The optimal exercise strategy attempts to determine the optimal the stopping time τ^* in order to maximize $V(t, S_t)$:

$$\tau^* = arg \max_{\tau \in \mathcal{T}_t} \mathrm{E}\left[e^{-r(\tau - t)}(K - S(\tau))^+ | \mathcal{F}_t\right].$$

The optimal stopping time coincides with the point where the stock price curve first intersects the optimal exercise boundary. The maximum value of the American option is attained. The optimal stopping time is defined as the first time when the price process hits this boundary,

$$\tau^* = \inf\left\{\tau \geq t : S_\tau = L^*(t)\right\},$$

where $L^*(t)$ is the optimal exercise boundary. i.e.,

Remark 2.3. When the time horizon T is infinite ($T = +\infty$), Problem 2.2 can be solved explicitly for perpetual American options. However, an explicit solution is not available for non-perpetual options with a finite expiry ($T < +\infty$), and numerical methods must be employed. This paper primarily focuses on the case where $T < +\infty$.

Remark 2.4. The characteristic of stock prices is similar to the Markov process. The future desired stock price depends only on the present stock price and is independent to the historical price. Hence

$$\mathrm{E}\left[S_q \mid \mathcal{F}_t\right] = \mathrm{E}\left[S_q \mid S_t\right], \quad \forall q \in [t, T].$$

When the stock curve intersects with the exercise boundary, the point of intersection denoted by τ is the optimal stopping time. Hence, it is necessary to first establish the optimal exercise boundary $L^*(t)$ to determine the optimal stopping time. Essentially, $L^*(t)$ divides the area into two regions, namely the Holding area and the Exercise area.

- Holding Area

$$\Sigma_1 = \{(S_t, t) \mid L^*(t) \leq S_t < \infty, 0 \leq t \leq T\} : V(S_t, t) > (K - S_t)^+.$$

- Exercise Area

$$\Sigma_2 = \{(S_t, t) \mid 0 \leq S_t < L^*(t), 0 \leq t \leq T\} : V(S_t, t) = (K - S_t)^+.$$

For the optimal control problem of American put options, the free boundary $L^*(t)$ is determined numerically together with the solution of the Black-Scholes equation. In order to characterize the optimal exercise policy, we need to find a solution pair $(V(S,t), L^*(t))$. Once the boundary is determined, the holders of the American put options make the exercise decisions based on the following rules:

$$\begin{cases} S_t \leqslant L^*(t), \text{ exercise the options} \\ L^*(t) < S_t \leq K, \text{ do not exercise the options and wait until the first hitting time} \\ S_t \geq K, \text{do not exercise the options} \end{cases}$$

We reformulate the optimal control problem of American options 2.2 into the following linear differential complementarity problem 2.6. By solving the linear differential complementarity problem, we can determine the solution of the optimal control problem 2.2.

The linear complementarity problem (LCP) is defined as follows.

Definition 2.5. (LCP)[16]. Let w be a mapping $w, \mathbb{R}^n \to \mathbb{R}^n$. Given w, a vector $z \in \mathbb{R}^n$ can be obtained as

$$w = Mz + q, \quad z \geq 0, w \geq 0, z_i w_i = 0, for \quad i = 1, 2, \ldots, n.$$

Problem 2.6. ***Linear differential complementarity problem of American options***

Determine V such that

$$\begin{cases} \mathcal{L}V \geq 0 \\ V - V^* \geq 0 \\ \mathcal{L}V \cdot (V - V^*) = 0 \end{cases}$$

in the region $\Omega := I \times (0,T)$, where $I = (0,X) \subset \mathcal{R}$ for a positive constant $X > K$. $\mathcal{L}$ denotes the Black-Scholes differential operator

$$\mathcal{L} := -\frac{\partial}{\partial t} - \frac{1}{2}\sigma^2 S^2 \frac{\partial^2}{\partial S^2} - rS\frac{\partial}{\partial S} + r, \tag{1}$$

where S denotes the asset price, $\sigma(t)$ is a known function characterizing the volatility of the asset, and $V^*$is the payoff function defined by $V^*(S) = \max\{K - S, 0\}$.

The final condition is given by

$$V(S,T) = V^*(S).$$

The boundary conditions are

$$V(0,t) = K, V(X,t) = 0.$$

Theorem 2.7. *Let $V(t, S_t)$ be the value of the American options, when the non-arbitrage assumption is implied. The American option pricing problem can be formally stated as the* ***linear differential complementarity problem in Problem 2.6****.*

Proof of Theorem 2.7. Consider the Delta-hedging portfolio of a long American option position and a short position in some quantity Δ_t of the stock

$$\Pi_t = V(S_t, t) - \Delta_t S_t. \tag{2}$$

The derivatives of (2) are

$$d\Pi_t = dV(S_t, t) - \Delta_t dS_t, \tag{3}$$

$$d\Pi_t = \frac{\partial V}{\partial t}dt + \frac{\partial V}{\partial S_t}dS_t + \frac{1}{2}\frac{\partial^2 V}{\partial S_t^2}dS_t - \Delta_t r S_t dt - \Delta_t \sigma S_t dW_t, \tag{4}$$

$$d\Pi_t = \left(\frac{\partial V}{\partial t} + rS_t\frac{\partial V}{\partial S_t} + \frac{1}{2}\sigma^2 S_t^2\frac{\partial^2 V}{\partial S_t^2}\right)dt + \sigma S_t\frac{\partial V}{\partial S_t}dw_t - \Delta_t rS_t dt - \Delta_t\sigma S_t dw_t.$$

To create a riskless portfolio, we choose $\Delta_t = \frac{\partial V}{\partial S}\left(S_t, t\right)$. The price movement of portfolio assets after the risk hedging is defined as

$$d\Pi_t = \left(\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S_t^2\frac{\partial^2 V}{\partial S_t^2}\right)dt.$$

- If $d\Pi_t > r\Pi_t dt$, the holder can arbitrage.
- If $d\Pi_t = r\Pi_t dt$, the holder has no chance to arbitrage. Hence,

$$V\left(S_t, t\right) > \left(k - S_t\right)^+,$$

$$\left(\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S_t^2\frac{\partial^2 V}{\partial S_t^2}\right)\cdot dt = r\left(V\left(t, S_t\right) - \frac{\partial V}{\partial S_t}\cdot S_t\right)dt.$$

- If $d\Pi_t < \Pi_t dt$, the holder has the possibility to arbitrage. If it is optimal to exercise, we do not have the arbitrage opportunity

$$V\left(S_t, t\right) = \left(k - S_t\right)^+,$$

$$\left(\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S_t^2\frac{\partial^2 V}{\partial S_t^2}\right)\cdot dt \leqslant r\left(V\left(t, S_t\right) - \frac{\partial V}{\partial S_t}\cdot S_t\right)dt.$$

The above analysis can be summarized as

$$\begin{cases} \frac{\partial V}{\partial t} + rS_t\frac{\partial V}{\partial S_t} + \frac{1}{2}\sigma^2 S_t^2\frac{\partial^2 V}{\partial S_t^2} - rV \leqslant 0 \\ V\left(S_t, t\right) - \left(k - S_t\right)^+ \geqslant 0 \\ \left(\frac{\partial V}{\partial t} + rS_t\frac{\partial V}{\partial S_t} + \frac{1}{2}\sigma^2 S_t^2\frac{\partial^2 V}{\partial S_t^2} - rV\right)\left(V\left(S_t, t\right) - \left(k - S_t\right)^+\right) = 0 \end{cases} \tag{5}$$

By defining $\mathcal{L}$ as equation (18), equation (5) can be simplified as

$$\begin{cases} \mathcal{L}V \geq 0 \\ V - V^* \geq 0 \\ \mathcal{L}V\cdot\left(V - V^*\right) = 0 \end{cases}$$

□

Determining the optimal exercise time for maximizing returns in American options is a more challenging task compared to European options. Indeed, the free boundary is the borderline of the holding region and the exercise region. Theoretical analysis proves that the graph of the free boundary is continuous and monotonically decreasing [18]. Remark 2.8 states the optimality conditions for the optimal exercise strategy. In the following, we abbreviate S_t as x.

Remark 2.8. Conditions for Optimal Exercise Strategy

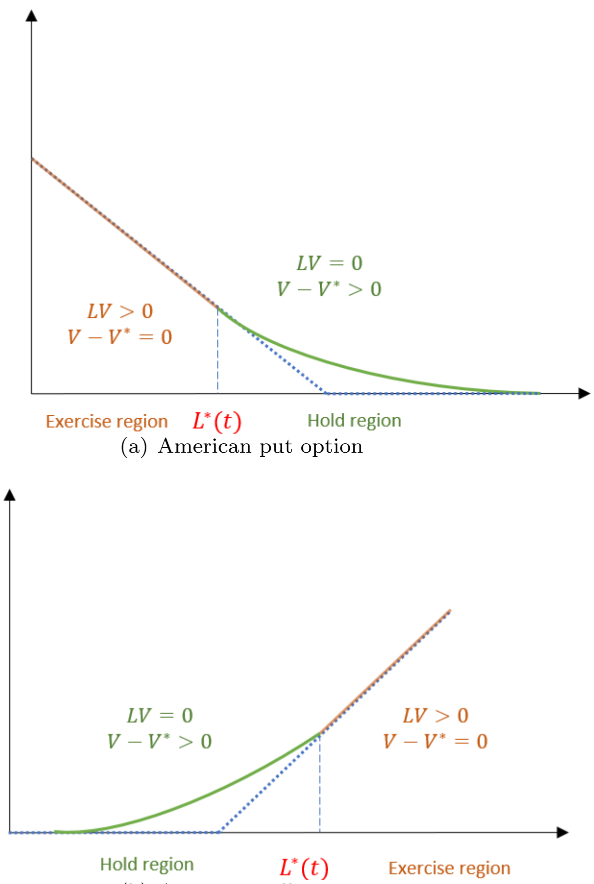
The free boundary and Optimal Exercise Strategy can be determined based on the difference between the expression of the option value in the holding region Σ_1 and the exercise region Σ_2,

If $\mathcal{L}V = 0$, $V - V^* > 0$, then $(x, t) \in \Sigma_1$ and the option is not exercised,

If $\mathcal{L}V > 0$, $V - V^* = 0$, then $(x, t) \in \Sigma_2$ and the option is exercised.

The free boundary is essentially the boundary which separates the holding region Σ_1 and the exercise region Σ_2.

Fig. 1 illustrates the determination of the exercise boundary for Remark 2.8. By analyzing the sign of $\mathcal{L}V$ and $V - V^*$ in relation to 0, the exercise and holding regions can be identified. Theorem 2.7 shows that the optimal control problem can be solved after the solution of the LCP problem is obtained. As depicted in Figure 2, the optimal exercise boundary is the optimal solution to the optimal control problem; the corresponding option value represents the optimal value.



(a) American put option

(b) American call option

FIGURE 1. Hold and exercise regions of American options

Remark 2.9. Wilmott [34] et al. proved that the American option pricing problem can be formulated as the following variational inequality:

$$\begin{aligned}
&\min\{-\mathcal{L}V, V - V^*\} = 0, \\
&V(x, T) = V^*(x) = \max\{K - x, 0\}, \\
&0 < t < T, (x, t) \in (0, X) \times (0, T).
\end{aligned} \tag{6}$$

The variational inequality (6) can also be solved by the proposed method in Section 3, where the proposed novel NN method is integrated with the penalty approach and the smoothing method to solve the linear differential complementarity problem in problem 2.6.

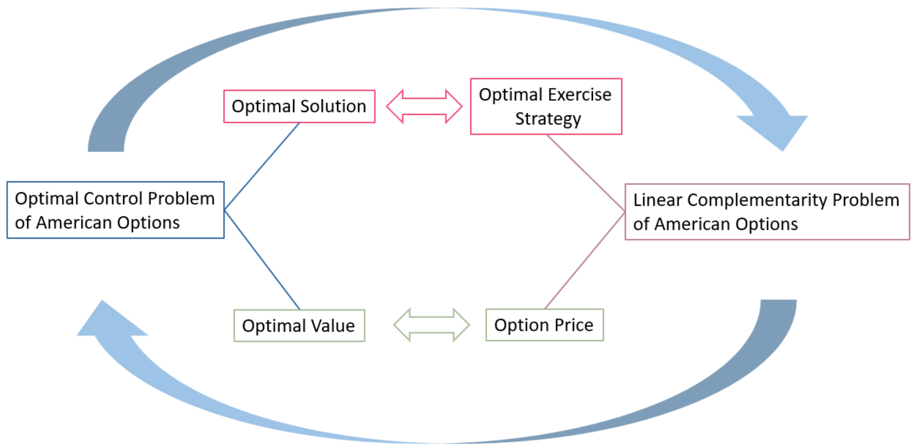


FIGURE 2. Problem Transformation

3. The novel NN method for the LCP.

3.1. Penalty approach and smoothing method. In this section, we first propose the penalty problem to approximate the *linear differential complementarity problem in Problem 2.6*, and then apply the smoothing method to process the non-differentiable part.

We define $[a]_+ = \max\{a, 0\}$, and then convert the two inequality constraint $\mathcal{L}V \geq 0$ and $V - V^* \geq 0$ into the nonlinear PDEs

$$\mathcal{L}V_\lambda(x,t) + \lambda \left[V^*(x) - V_\lambda(x,t)\right]_+^{1/k} = 0,$$

with the boundary and final conditions

$$V_\lambda(x,T) = V_\lambda^*(x),\, V_\lambda(0,t) = K, V_\lambda(X,t) = 0,$$

where λ is a penalty parameter which is a reasonably large number.

The equations of the original LCP problem can be approximated by the following PDEs of the bi-nolinear system

$$\mathcal{L}V_\lambda(x,t) + \lambda \left[V^*(x) - V_\lambda(x,t)\right]_+^{1/k} = 0, \tag{7}$$

$$\mathcal{L}V_\lambda \cdot (V_\lambda - V^*) = 0. \tag{8}$$

Remark 3.1. The term $[V^*(x) - V_\lambda(x,t)]_+^{1/k}$ is mainly used to penalize the positive part of $V^*(x) - V_\lambda(x,t)$. If $[V^*(x) - V_\lambda(x,t)]_+$ is equal to 0, the two inequalities $V - V^* \geq 0$ and $\mathcal{L}V \geq 0$ are satisfied. If $[V^*(x) - V_\lambda(x,t)]_+$ is not equal to 0, then the inequality $V - V^* \geq 0$ is not satisfied. If $[V^* - V_\lambda]_+ = \lambda^{-k} \left(-\mathcal{L}V_\lambda\right)^k$, it can guarantee that $[V^* - V_\lambda]_+ \approx 0$ when λ is large enough.

Problem 3.2. *Determine* $V_\lambda(x,t)$ *for all* $(x,t) \in \Omega := (0,T) \times I$,

$$\begin{aligned}
&\mathcal{L}V_\lambda(x,t) + \lambda\,[V^*(x) - V_\lambda(x,t)]_+^{1/k} = 0,\\
&\mathcal{L}V_\lambda \cdot (V_\lambda - V^*) = 0,\\
&V_\lambda(x,T) = V^*(x) = \max\{K-x,0\},\\
&V_\lambda(0,t) = K,\\
&V_\lambda(X,t) = 0,
\end{aligned} \tag{9}$$

where λ *is the penalty parameter.*

Problem 3.2 has a unique solution, where we refer to [32] for proof of convergence.

The red line in Figure 3 shows that $V_\lambda(x)$ is not differentiable at the point $x = K$. Therefore, the smoothing technology is applied on the non-smooth function $V_\lambda(x) = \max\{K - x, 0\}$ in the ε neighborhood of the unsmooth angle point $x = K$. The function V is approximated as the smooth function V_λ^ε.

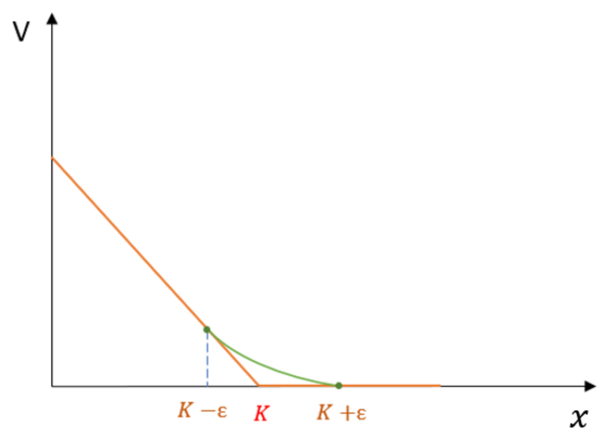


FIGURE 3. Smoothing method

$$V_\lambda^\varepsilon(x) := \begin{cases} V_\lambda, & x < K-\varepsilon,\\ \frac{(V_\lambda-\varepsilon)^2}{4}, & K-\varepsilon \le x \le K+\varepsilon,\\ 0, & x > K+\varepsilon. \end{cases} \tag{10}$$

We refer to [29] for proof of convergence.

3.2. **A novel neural networks algorithm.** Unlike the traditional data-driven methods, we propose a novel NN by incorporating the structural properties of PDE with the training process. By minimizing a loss function, the neural network attempts to model the PDE. Additionally, known data points can be easily incorporated into the loss function based on the structural properties of the PDE in order to facilitate faster training and enhance the network performance. To solve the bi-nolinear system of PDEs, we apply the simplest feed-forward neural network method.

3.2.1. *Feed-forward neural neworks (FNNs).* FNNs are the first and simplest artificial neural network [26]. Each layer of the FNN consists of multiple neurons, where information flows exclusively from the input node to the output node through the hidden node. This structural design enables FNNs to effectively handle sequence data and time series tasks. As depicted in Fig. 4, each neuron receives signals from neurons in the previous layer and generates outputs for the subsequent layer. By incorporating the activation functions, FNNs are capable of modeling nonlinearity and facilitating mapping tasks. In theory, it can approximate any continuous function [11]. V_θ in equation (11) is the neural network approximation of the solution

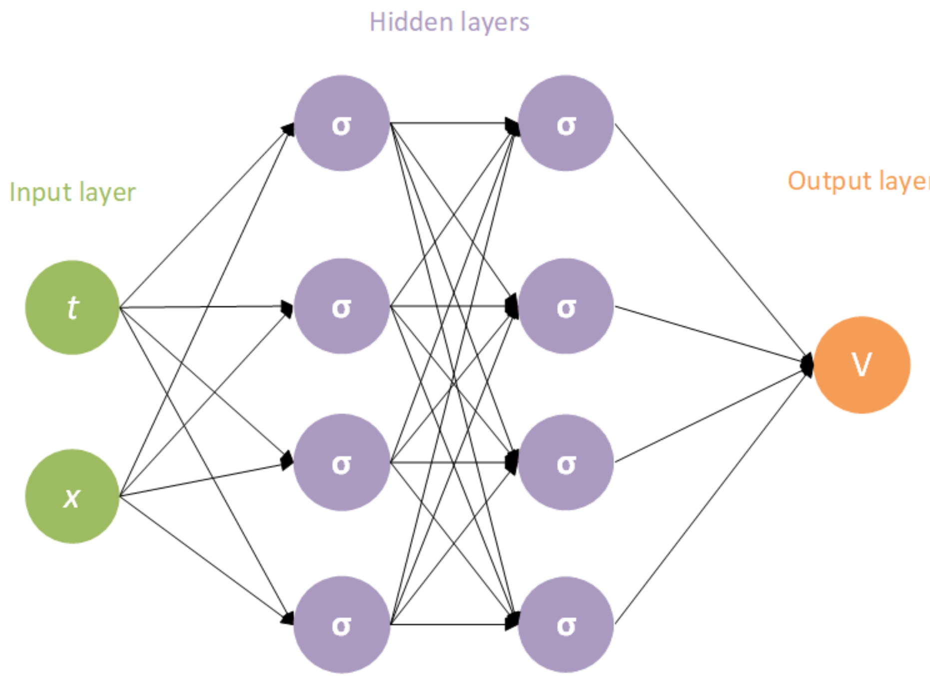


FIGURE 4. Feed-Forward Neural Nework

V.

$$V_\theta(z) := W^L\sigma^L\left(W^{L-1}\sigma^{L-1}\left(\cdots\sigma^1\left(W^0z+b^0\right)\cdots\right)+b^{L-1}\right)+b^L, \tag{11}$$

where W^L and b^L are the weight matrices and bias vectors, and $z=[x,t]^T$. The activation function σ plays a significant role in neural networks. Nonlinear activation functions are generally used to model nonlinear components. Therefore, the neural network is able to approximate a wide range of nonlinear functions. There are various popular activation functions utilized in neural networks, such as the Sigmoid, ReLU, and Tanh functions.

The training of neural networks often involves the determination of the derivatives of V with respect to the unknown parameters W^L and b^L. Gradient-based optimization methods such as gradient descent are generally used to determine the derivatives [6]. To simplify the gradient computation, automatic differentiation (AD) techniques implemented in TensorFlow [27] or PyTorch [22] are widely used. The AD techniques automatically compute the derivatives of the neural network with respect to its inputs and parameters by the chain rule. The approach avoids manual gradient calculations. Hence, the training process is more efficient, especially when solving the PDEs.

As illustrated in Figure 5, the error function of the NN method is determined by both the data points and PDEs. Indeed, the structural properties of PDEs are added as the constraints to the neural network in order to ensure that the training results satisfy the characteristics of the PDEs.

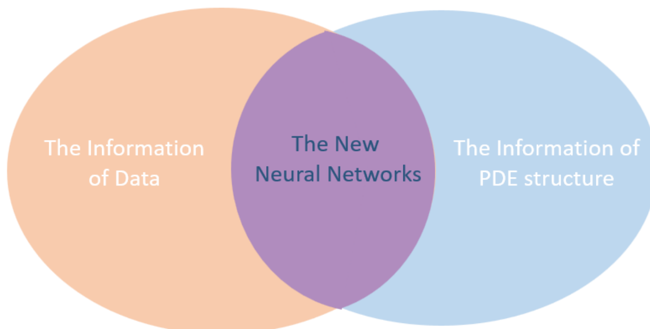


FIGURE 5. The error function of novel NNs relies on both data and the PDE

The loss function ϕ_θ is formulated with the following three terms:

1. The mean squared error quantifies the averaged difference between the initial and terminal conditions in a number of collocation points,
$$\begin{aligned}\phi_\theta^0\left(X^0\right) &:= \frac{1}{N_0}\sum_{i=1}^{N_0}\left|u_\theta\left(x_i^0,t_i^0\right)-u_0\left(x_i^0,t_i^0\right)\right|^2,\\ \phi_\theta^f\left(X^f\right) &:= \frac{1}{N_f}\sum_{i=1}^{N_f}\left|u_\theta\left(x_i^f,t_i^f\right)-u_f\left(x_i^f,t_i^f\right)\right|^2,\end{aligned}\tag{12}$$
where
$$X^0 := \left\{\left(x_i^0,t_i^0\right)\right\}_{i=1}^{N_0}\subset I\times 0,\ \text{and } X^f := \left\{\left(x_i^f,t_i^f\right)\right\}_{i=1}^{N_f}\subset I\times T,\tag{13}$$
and X^0 and X^f denote the initial and termination points. The two equations in (12) represent the mean squared error between the neural network approximates and the true values at the initial and termination points, respectively.
2. The mean squared error quantifies the averaged difference between the upper and lower boundary conditions in a number of collocation points,
$$\begin{aligned}\phi_\theta^{ub}\left(X^{ub}\right) &: = \frac{1}{N_{ub}}\sum_{i=1}^{N_{ub}}\left|u_\theta\left(x_i^{ub},t_i^{ub}\right)-u_{ub}\left(x_i^{ub},t_i^{ub}\right)\right|^2,\\ \phi_\theta^{lb}\left(X^{lb}\right) &: = \frac{1}{N_{lb}}\sum_{i=1}^{N_{lb}}\left|u_\theta\left(x_i^{lb},t_i^{lb}\right)-u_{lb}\left(x_i^{lb},t_i^{lb}\right)\right|^2,\end{aligned}\tag{14}$$
$$X^{ub} := \left\{\left(x_i^{ub},t_i^{ub}\right)\right\}_{i=1}^{N_{ub}}\subset 0\times[0,T],\ \text{and } X^{lb} := \left\{\left(x_i^{lb},t_i^{lb}\right)\right\}_{i=1}^{N_{lb}}\subset X\times[0,T],\tag{15}$$
where X^{ub} and X^{lb} represent the upper and lower boundary points. The two equations in (14) represent the mean squared errors, which quantify the

differences between the function u approximated by the neural network and the true value at upper and lower boundary points, respectively.

3. The mean squared error of inner allocation point

$$\phi_\theta^r\left(X^r\right) := \frac{1}{N_r}\sum_{i=1}^{N_r}\left(\left|r_\theta^1\left(x_i^r, t_i^r\right)\right|^2 + \left|r_\theta^2\left(x_i^r, t_i^r\right)\right|^2\right), \tag{16}$$

in which $X^r := \{(x_i^r, t_i^r)\}_{i=1}^{N_r} \subset I \times (0, T)$, and $r_\theta^1(x,t)$ and $r_\theta^2(x,t)$ denote the residuals of the bi-nonlinear PDEs which are given as

$$\begin{aligned} r_\theta^1(x,t) &:= \mathcal{L}V_\lambda(x,t) + \lambda\left[V^*(x) - V_\lambda(x,t)\right]_+^{1/k}, \\ r_\theta^2(x,t) &:= \mathcal{L}V_\lambda \cdot \left(V_\lambda - V^*\right), \end{aligned} \tag{17}$$

where

$$\mathcal{L} := -\frac{\partial}{\partial t} - \frac{1}{2}\sigma^2 x^2 \frac{\partial^2}{\partial x^2} - rx\frac{\partial}{\partial x} + r. \tag{18}$$

Incorporating the above three errors, the unknown parameters of the NN are trained by minimizing the loss function

$$\phi_\theta(X) := \phi_\theta^r\left(X^r\right) + \phi_\theta^0\left(X^0\right) + \phi_\theta^f\left(X^f\right) + \phi_\theta^{ub}\left(X^{ub}\right) + \phi_\theta^{lb}\left(X^{lb}\right), \tag{19}$$

where X denotes the collection of training data including the inner points X^r, the initial points X^0, the termination points X^f, and the boundary points X^{lb} and X^{ub}.

3.2.2. *Numerical algorithm for optimal control problems.* Algorithm 1 demonstrates the utilization of the novel NN method to solve the LCP. This method relies mainly on the scientific computing library NumPy and the machine learning library TensorFlow. The process of determining the numerical solution to the optimal control problem is summarized in Figure 6. Initially, we convert the optimal control problem associated with American options into a LCP. We then employ a penalty approach and smoothing method to transform the problem as a solution for a bi-nonlinear PDE. Consequently, we solve this PDE using the the novel NN method. By combining the structural properties of the PDE equation and the information of the sampling points, we evaluate the the overall mean squared error in (19). The unknown parameters in the neural networks are trained to approximate the solution of the LCP. The numerical solution to the optimal control problem of American options can be obtained.

Here is the neural network configuration which was used to solve this LCP:

- Scale the input in the interval [-1,1];
- Fully connected layers;
- Activation function: *Tanh*, $\sigma(x) = tanh(x) = \frac{2}{1+e^{-2x}} - 1$;
- Number of layers, $L = 2$ or 4;
- Number of neurons in the neural network, $D = 4$ or 8.

Remark 3.3. There is no fixed rule to determine the optimal numbers of hidden nodes and layers in a neural network, which depend on the complexity and landscape of the problem being addressed. Hence, the trial and error method is used to determine the optimal neural network configuration.

Algorithm 1 Solving the LCP by the novel NN method

1: **Input**: Number of layers, L; Number of neurons in the neural network, D; Maximum number of training iterations, I_{max}; The learning rate, lr;
2: **Output**: Option value: V;
3: Generate a set of collocation points. Generate the inner points X^r, the initial points X^0, the boundary points X^{ub} and X^{lb} and the finial points X^f by uniformly distributed random number generations.
4: Determine the gradient and loss values:
- Compute the partial derivatives $\frac{\partial V}{\partial t}$, $\frac{\partial V}{\partial x}$, $\frac{\partial V^2}{\partial t^2}$, and $\frac{\partial V^2}{\partial x^2}$ by automatic differentiation (AD) capabilities;
- Compute the mean squared residual $\phi_\theta^r(X^r)$, $\phi_\theta^0(X^0)$, $\phi_\theta^f(X^f)$, $\phi_\theta^{ub}(X^{ub})$, and $\phi_\theta^{lb}(X^{lb})$, by equations (16),(12), (13), and (14).
- Evaluate the LCP residual by equation (19).

5: Set up the optimizer and train the NN:
- Set the learning rate lr to the step function which decays in a piecewise constant fashion;
- Update the NN weights using the Adam optimizer.

6: **Return**: The optimized parameters W^L, b^L of the NN, and the option value V.

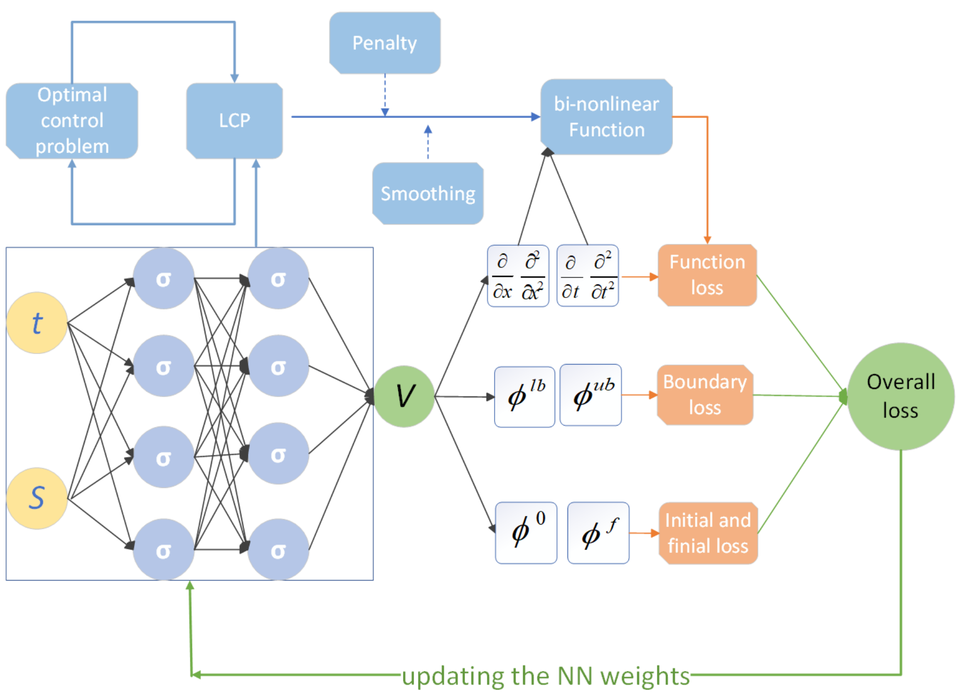


FIGURE 6. Flow of solving the optimal control problem of American options

4. **Numerical results.** In this section, we present three numerical examples to demonstrate the efficiency and usefulness of the proposed NN method.

In the first example, we demonstrate the effectiveness of the proposed NN method to solve the LCP. We consider a test LCP example with unsymmetric system matrices. By comparing the numerical solutions obtained by the proposed NN method and those presented in Test 1 of Wang's article [33], we can confirm that the proposed NN is an effective approach to solve the LCP. For the second and third examples, we solve the optimal control problems for American call and put options, respectively. The results are compared with those presented in [31] and [32]. The effectiveness of the proposed NN method-based algorithm for solving the American option optimal control problem can be further demonstrated. The proposed NN method is implemented in a Python environment with TensorFlow. All the computations are performed on a Macbook Pro equipped with a Apple M1 Max-Core processor and 64G of RAM.

4.1. **Example 1.** The LCP with un-symmetric system matrices in [33]

$$\begin{aligned} \mathcal{L}_1 u &\leq f, \\ \mathcal{L}_2 u &\leq u^*, \\ (\mathcal{L}_1 u - f) \cdot (\mathcal{L}_2 u - u^*) &= 0, \end{aligned} \tag{20}$$

is considered, where

$$\begin{aligned} \mathcal{L}_1 &= -\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right) + \frac{\partial}{\partial x} + \frac{\partial}{\partial y}, \mathcal{L}_2 = I, \\ f &= 5\left[6xy\left(2 - x^2 - y^2\right) + \left(1 - 3x^2\right)\left(y - y^3\right) + \left(1 - 3y^2\right)\left(x - x^3\right)\right], \\ u^* &= 0.3 + |x - 0.5| + |y - 0.5|, \end{aligned} \tag{21}$$

$(x, y) \in \Omega := (0, 1) \times (0, 1)$, $u = 0$ on the boundary of Ω, and I denotes the identity operator. For the unconstrained condition $\mathcal{L}_1 u \leq f$, the exact solution of $\mathcal{L}_1 u = f$ is $u = 5(x - x^3)(y - y^3)$.

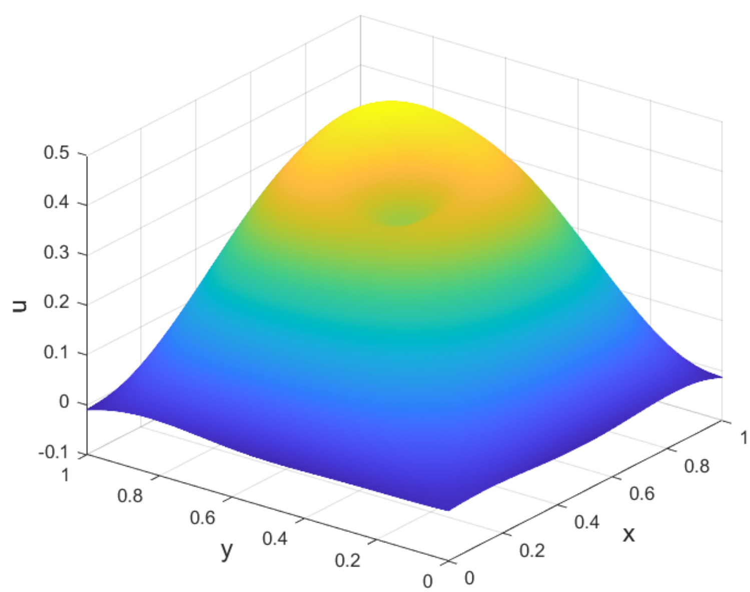


FIGURE 7. The value of u

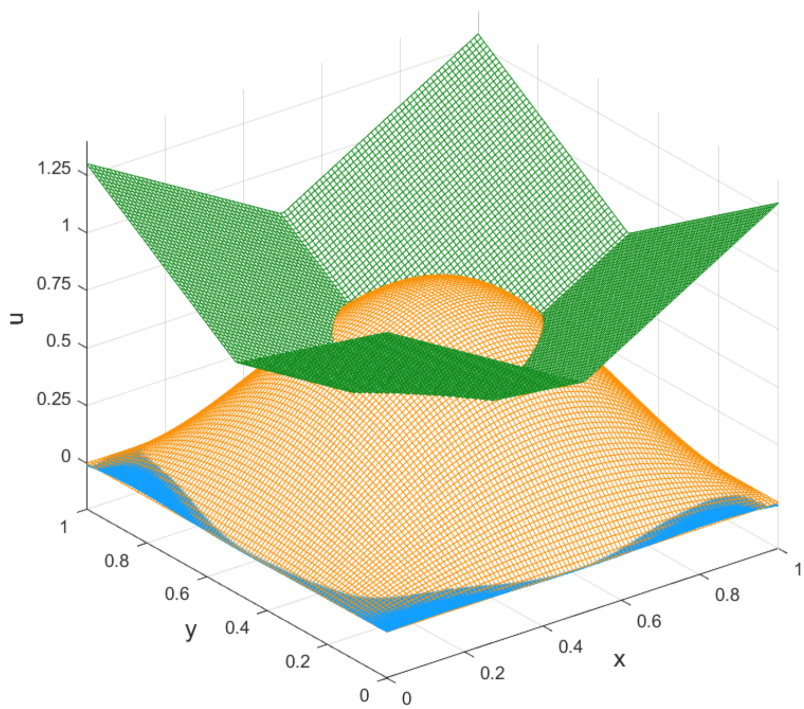


FIGURE 8. The value of u and constraint conditions

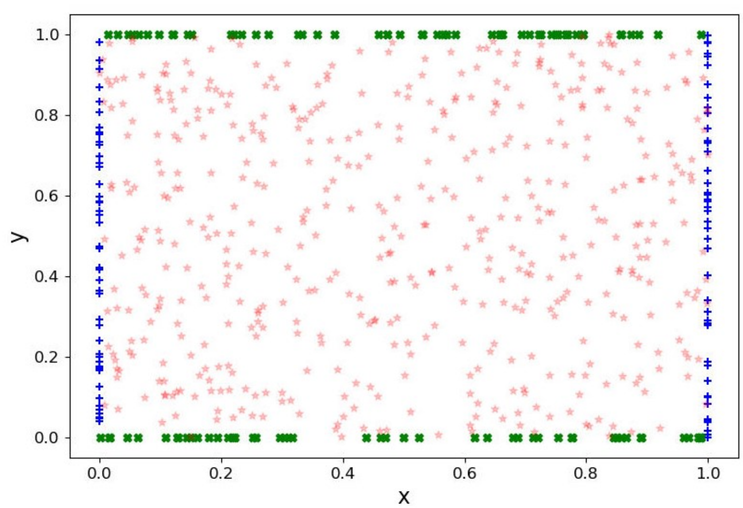


FIGURE 9. Positions of collocation points

We solve this problem by the proposed method. After performing several trials, we found that the neural network with 4 hidden layers and 8 neurons in each hidden layer is able to achieve better results. To optimize the learning process, we

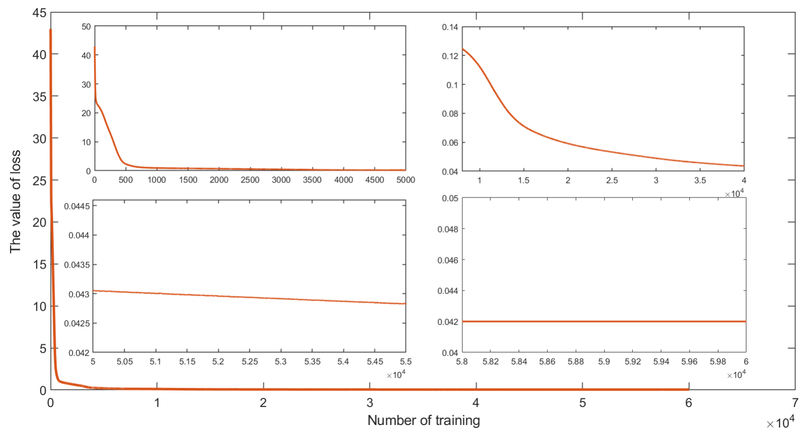


FIGURE 10. The convergence plots for the loss function

choose a piecewise decay of the learning rate. Specifically, the learning rate was set to 0.001 for the first 7000 steps, followed by a rate of 0.0001 from step 7000 to 40,000, and finally 0.00001 from step 40,000 onwards. Figure 7 shows the results of the numerical solution u. To further demonstrate the effectiveness of addressing the inequality constraints, we plot the planes corresponding to the two inequality equations separately in Figure 8. The green and orange surfaces represent $\mathcal{L}_1 u = f$ and $\mathcal{L}_2 u = u^*$, respectively. The blue surfaces represent the numerical solution of u. Notably, the blue region is consistently below the green and orange sub-planes. This result validates that u satisfies the inequality constraint of the LCP.

By comparing the calculation results with Test 1 in Wang's article [33], the numerical solution is consistent with the solution of Wang's article. This consistency indicates that the proposed Novel NN method is able to solve the LCP accurately. In contrast, the existing method proposed in [33] requires a complex transformation of the LCP to an alternative form; this complex transformation can be omitted by the proposed NN. This complex transformation is essential for the existing method and requires ongoing modification with different LCP. Hence, it is apparent that the existing method has no generalizability [33].

As a combination of data-driven and structural properties-driven methods, the quality of the dataset plays a critical role in the success of training. The results of training are also influenced by the dataset's accuracy on each constraint boundary. In this example, we use the built-in function $tf.random.uniform()$ in the TensorFlow library to generate a specified number of random points in a uniform distribution. To satisfy the boundary condition $u = 0$ on the boundary of Ω, we choose 100 points on each of the top, bottom, left, and right boundaries. Additionally, we randomly select 500 points within the interior of Ω with a uniform distribution. Figure 9 shows the distribution of the randomly sampled points, while Figure 10 shows the convergence curve of the cost function. To emphasize the reduction in the value function, we present enlarged results of certain iteration steps as subplots within Figure 10. The first three subplots clearly show the loss values decrease significantly. The last subplot shows that the loss values converge to the order of 1e-2 when the number of iterations increases.

4.2. **Example 2.** In this example, we solve the American call option test problem in [31] by the proposed NN. For American call options, it is not wise to exercise the option early. Hence, the optimal exercise time is the termination moment, and determination of the option value is necessary.

Call option: Let V denote the value of the call option, and x the price of the underlying asset. V satisfies the Black–Scholes equation

$$LV := -\frac{\partial V}{\partial t} - \frac{1}{2}\sigma^2(t)x^2\frac{\partial^2 V}{\partial x^2} - (r(t)x - D(x,t))\frac{\partial V}{\partial x} + rV = 0, \tag{22}$$

where $(x,t) \in I \times [0,T)$ with $I = [0,X]$. The boundary conditions and the final condition are given as (23) and (24), respectively.

$$\begin{aligned} &V(0,t) = 0, \\ &V(X,t) = X\exp\left(-\int_t^T d(X,\tau)\mathrm{d}\tau\right) - E\exp\left(-\int_t^T r(\tau)\mathrm{d}\tau\right), \end{aligned} \tag{23}$$

$$V(x,T) = \max(0, x-E), \quad x \in \bar{I}. \tag{24}$$

The parameters are set as, $X = 700, T = 1, r = 0.1, \sigma = 0.3, d = 0.04$, and $E = 400$.

The NN has two hidden layers, and the numbers of hidden nodes in the two layers are 2 and 4. We choose a piecewise decay of the learning rate. The first 10,000 steps use a learning rate of 0.05; a learning rate of 0.005 is used from steps 10,000 to 25,000; a learning rate of 0.001 is used from steps 25,000 to 40,000; and a learning rate of 0.0005 is used from step 40,000 onwards.

In this example, we use the built-in function $tf.random.uniform()$ in the Tensorflow library to generate a specified number of random points in a uniform distribution. Given the boundary conditions in (23) and the final condition in (24), 50 points are distributed at the left and right boundaries of Ω, 200 points are distributed at the top boundary of Ω, and 2000 points are distributed in the interior of Ω with uniform randomness. Figure 11 shows the stochastic distribution of the collocation points.

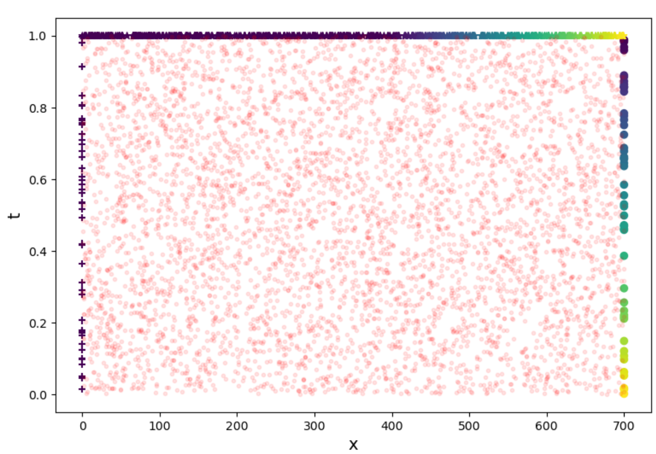


FIGURE 11. Positions of collocation points

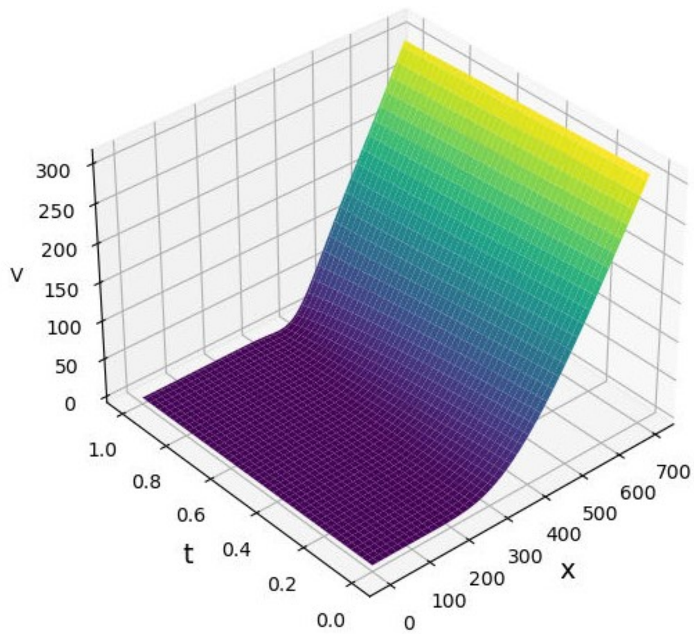


FIGURE 12. The value of the option value V

Figure 12 shows the value of the option value V. By comparing the existing result which is illustrated in Test 1 in [31], it was found that the results obtained by the proposed novel NN method are consistent with the existing results. However, the advantage of the novel NN method is that it does not require the complicated process which has been implemented in discretization and transformation of the problem. Additionally, we can solve this more complicated American call option test problem with minimal modifications to the program which has been implemented to solve the simpler problem in Example 1. Therefore, we can solve similar structured problems without significant modification to the original program, although solving the more complicated problem is required.

4.3. **Example 3.** In this example, we solve the LCP which is transformed by the optimal control problem of American put option. The following parameters are used: $X = 100, T = 1.5, r = 0.03, \sigma = 0.4$, and $K = 50$.

In order to solve this problem by the proposed method, 80 random points are distributed on both the upper and lower boundaries. Additionally, 80 and 500 points are selected for the final region and the middle region respectively. Figure 13 illustrates that the points are randomly distributed on the boundaries and in the middle region.

After several trials, better results can be achieved by setting the penalty parameter as $\lambda = -2$ and the smoothing coefficient as $\varepsilon = 6$. By applying these values, the transformed LCP for Problem 3.2 can be solved by using the proposed numerical method efficiently. After performing many trials, we found that the proposed neural network is able to achieve better results when the following neural network configuration and learning rates are used: 8 neurons per layer and 4 hidden layers,

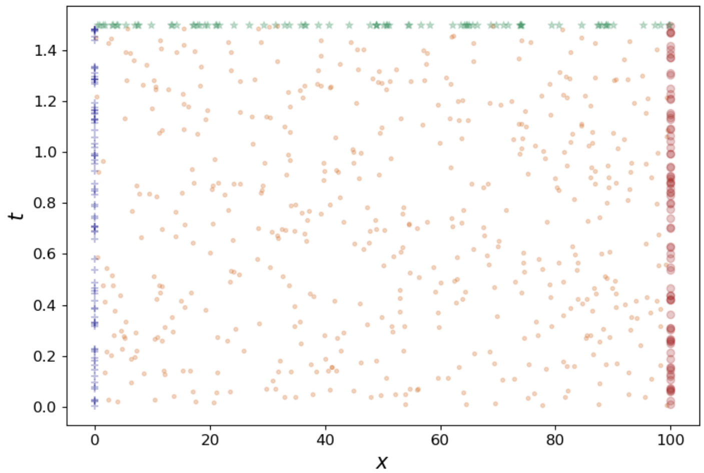


FIGURE 13. American put option value

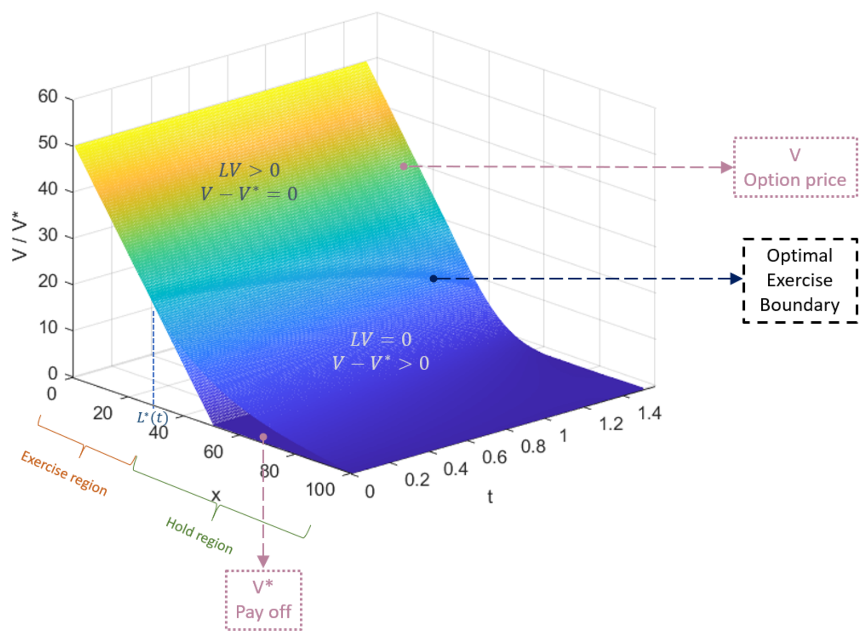


FIGURE 14. American put option value

learning rate of 0.001 for the first 15000 steps, a learning rate of 0.0001 from steps 15,000 to 35,000, and a learning rate of 0.0001 from step 35,000 onwards.

Figures 14 and 15 show the value of the American put option price, which is the optimal value of the optimal control problem. Figure 14 includes the option value along with the payoff, which can be used to identify the exercise boundary by

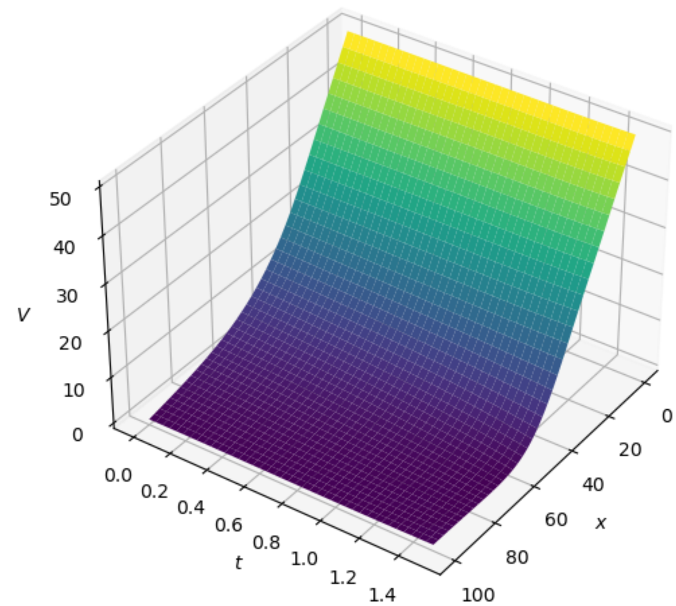


FIGURE 15. American put option value/optimal values of the optimal control problem

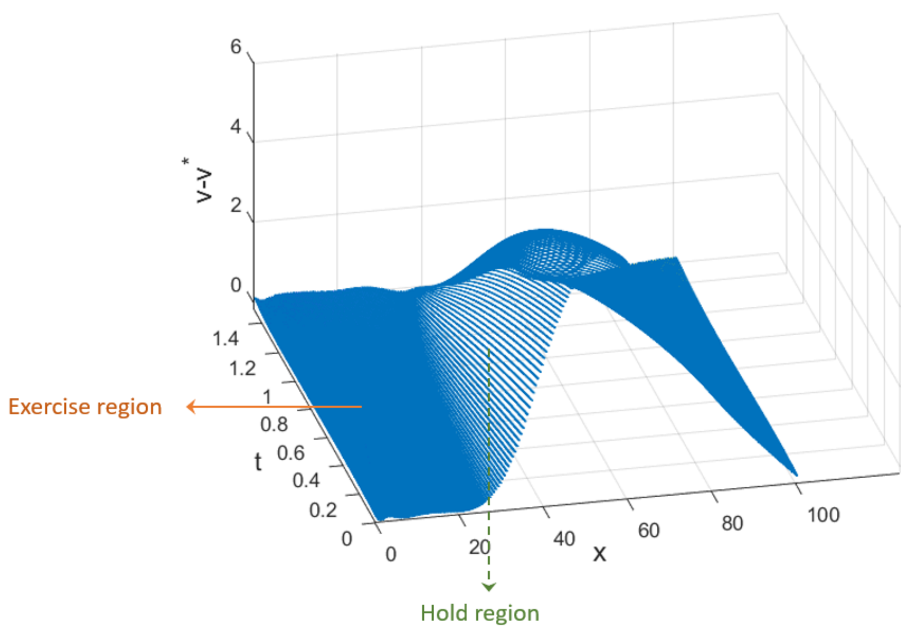


FIGURE 16. The value of $V - V^*$

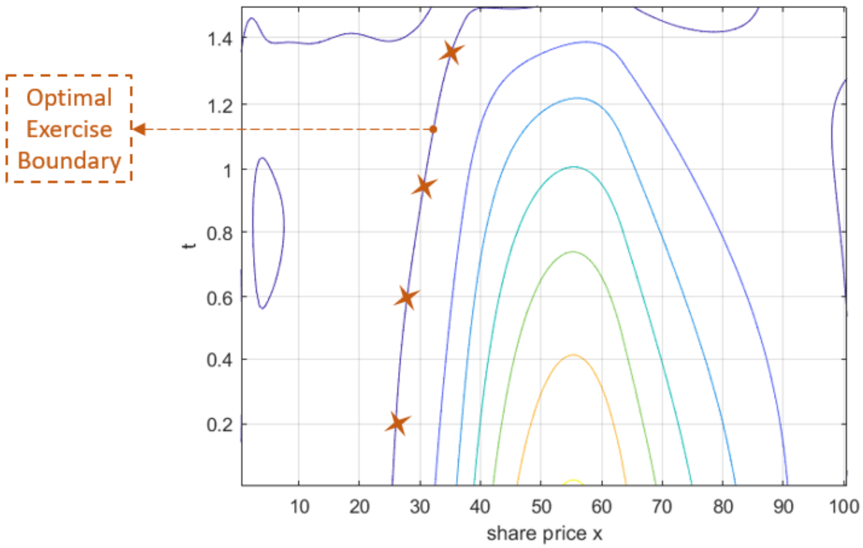


FIGURE 17. The contour view of $V - V^*$

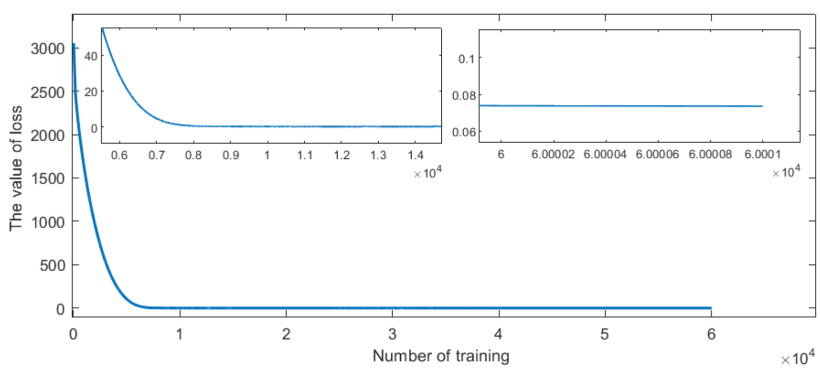


FIGURE 18. The convergence plots for the loss function

observing the intersection line. For this optimal control problem, this intersection line indicates the optimal solution. By performing the computation in the regions on both sides of the line, it can be found that the equality $V - V^* = 0$ is satisfactory within the left region when $\mathcal{L}V > 0$. In the right region, $\mathcal{L}V = 0$, $V - V^* > 0$ is satisfactory. Figure 16 presents the value of $V - V^*$, where the solid part represents the holding area and the dotted part represents the exercise area. According to Condition 2.8, the intersection of the holding area and exercise area corresponds to the optimal exercise boundary. In addition, the contour view of $V - V^*$ is presented in Figure 17 to show the optimal exercise boundary, where the star sign denoting the optimal exercise boundary is consistent with that of Figure 14. Furthermore, Figure 18 shows the values of the loss function. When the training process progresses, the value of the loss function decreases and converges to the order of 10e-2. The two upper subplots in Figure 18 provide an enlarged view of the certain iteration steps. The upper left subplot shows the decreasing trend more clearly, while the upper right subplot shows that the loss values converge to the order of 1e-2 when the number of iteration steps increases.

5. **Conclusion.** In conclusion, the proposed novel NN method offers a valuable and practical approach to solving the American option optimal control problem. By reformulating the problem as the solution of PDEs in the bi-nonlinear system of PDEs and employing neural networks to approximate the PDE solutions, we have demonstrated the effectiveness of our method in providing optimal exercise strategies and option values for American options. This method on the financial market is significant as it offers a more efficient and accurate way to solve American option problems compared to existing methods. This can lead to better decision-making in financial trading and risk management, ultimately contributing to improved market efficiency and stability. Looking ahead, future research can explore the application of the novel NN method to other financial derivatives and risk management problems. Additionally, potential improvements in the method, such as incorporating additional market factors or refining the neural network architecture, can further enhance its applicability and accuracy in real-world financial situations.

Acknowledgments. The paper is supported by the Hong Kong Polytechnic University Grants (4-ZZPT, 1-WZ0E) and Research Centre for Quantitative Finance Grant (1-CE28).

Data availability. We do not analyse or generate any datasets, because our work proceeds within a theoretical and mathematical approach. One can obtain the relevant materials from the references below.

Declarations. Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this paper.
Consent for publication This is to state that the authors give full permission for the publication of the article.

REFERENCES

[1] D. Anderson and U. Ulrych, Accelerated American option pricing with deep neural networks, *Quantitative Finance and Economics*, **7** (2023), 207-228.

[2] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria and U. R. Acharya, ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis, *Future Generation Computer Systems*, **115** (2021), 279-294.

[3] R. Bellman, Dynamic programming, *Science*, **153** (1966), 34-37.

[4] J. Blechschmidt and O. G. Ernst, Three ways to solve partial differential equations with neural networks-A review, *GAMM-Mitteilungen*, **44** (2021), e202100006.

[5] M. J. Brennan and E. S. Schwartz, The valuation of American put options, *The Journal of Finance*, **32** (1977), 449-462.

[6] L. Bottou, F. E. Curtis and J. Nocedal, Optimization methods for large-scale machine learning, *SIAM Review,* **60** (2018), 223-311.

[7] P. P. Boyle, Options: A monte carlo approach, *Journal of Financial Economics*, **4** (1977), 323-338.

[8] B. Chen and P. T. Harker, A non-interior-point continuation method for linear complementarity problems, *SIAM Journal on Matrix Analysis and Applications*, **14** (1993), 1168-1190.

[9] C. Chen, H. Modares, K. Xie, F. L. Lewis, Y. Wan and S. Xie, Reinforcement learning-based adaptive optimal exponential tracking control of linear systems with unknown dynamics, *IEEE Transactions on Automatic Control*, **64** (2019), 4423-4438.

[10] J. C. Cox, S. A. Ross and M. Rubinstein Option pricing: A simplified approach, *Journal of Financial Economics*, **7** (1979), 229-263.

[11] B. C. Csáji, Approximation with artificial neural networks, *Faculty of Sciences, Etvs Lornd University, Hungary*, **24** (2001), 7.

[12] S. P. Dirkse and M. C. Ferris, The path solver: A nommonotone stabilization scheme for mixed complementarity problems, *Optimization Methods and Software*, **5** (1995), 123-156.

[13] R. M. Gaspar, S. D. Lopes and B. Sequeira, Neural network pricing of American put options, *Risks*, **8** (2020), 73.

[14] S. Gu, B. Kelly and D. Xiu, Empirical asset pricing via machine learning, *The Review of Financial Studies*, **33** (2020), 2223-2273.

[15] E. Haghighat, M. Raissi, A. Moure, et al., A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics, *Computer Methods in Applied Mechanics and Engineering*, **379** (2021), 113741.

[16] P. T. Harker and J.-S. Pang, For the linear complementarity problem, *Lectures in Applied Mathematics*, **26** (1990), 265-284.

[17] T. J. R. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Courier Corporation, 2012.

[18] L. Jiang and C. Li, *Mathematical Modeling and Methods of Option Pricing*, World Scientific, 2005.

[19] J. J. Júdice and A. M. Faustino, A computational analysis of LCP methods for bilinear and concave quadratic programming, *Computers & Operations Research*, **18** (1991), 645-654.

[20] F. L. Lewis, D. L. Vrabie and V. L. Syrmos, *Optimal Control*, John Wiley & Sons, 2012.

[21] X. Ma, Y. Niu, L. Gu, et al., Understanding adversarial attacks on deep learning based medical image analysis systems, *Pattern Recognition*, **110** (2021), 107332.

[22] A. Paszke, S. Gross, F. Massa, et al., Pytorch: An imperative style, high-performance deep learning library, *Advances in Neural Information Processing Systems*, **32** (2019).

[23] L. S. Pontryagin, *Mathematical Theory of Optimal Processes*, Routledge, 2018.

[24] M. Raissi, P. Perdikaris and G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *Journal of Computational Physics,* **378** (2019), 686-707.

[25] B. Salvador, C. W. Oosterlee and R. van der Meer, Financial option valuation by unsupervised learning with artificial neural networks, *Mathematics*, **9** (2020), 46.

[26] J. Schmidhuber, Deep learning in neural networks: An overview, *Neural Networks*, **61** (2015), 85-117.

[27] N. Shukla and K. Fricklas, *Machine Learning with TensorFlow*, Greenwich, Manning, 2018.

[28] P. R. Takács and Z. Darvay, A primal-dual interior-point algorithm for symmetric optimization based on a new method for finding search directions, *Optimization*, **67** (2018), 889-905.

[29] K. L. Teo, B. Li, C. Yu, et al., *Applied and Computational Optimal Control*, Optimization and Its Applications, 2021.

[30] N. Wang, Y. Gao, H. Zhao and C. K. Ahn, Reinforcement learning-based optimal tracking control of an unknown unmanned surface vehicle, *IEEE Transactions on Neural Networks and Learning Systems*, **32** (2020), 3034-3045.

[31] S. Wang, A novel fitted finite volume method for the Black-Scholes equation governing option pricing, *IMA Journal of Numerical Analysis*, **24** (2004), 699-720.

[32] S. Wang and X. Yang, A power penalty method for a bounded nonlinear complementarity problem, *Optimization*, **64** (2015), 2377-2394.

[33] S. Wang and K. Zhang, An interior penalty method for a finite-dimensional linear complementarity problem in financial engineering, *Optimization Letters*, **12** (2018), 1161-1178.

[34] P. Wilmott, J. Dewynne and S. Howison, Option pricing: Mathematical models and computation, (1993).

[35] S.-L. Wu and C.-X. Li, A generalized Newton method for non-Hermitian positive definite linear complementarity problem, *Calcolo*, **54** (2017), 43-56.

[36] Y. Xu, J. Du, L. R. Dai and C. H. Lee, An experimental study on speech enhancement based on deep neural networks, *IEEE Signal Processing Letters*, **21** (2013), 65-68.

[37] K. Zhang, H. Song and J. Li, Front-fixing FEMs for the pricing of American options based on a PML technique, *Applicable Analysis*, **94** (2015), 903-931.

Received November 2023; revised March 2024; early access June 2024.