Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 8s (2025)

Deep Learning for Non-Linear Black-Scholes Model in an Illiquid Financial Market with Transaction Costs

Tejal Shah[1], **Jaita Sharma**[2]

[1]Department of Applied Mathematics, Faculty of Technology and Engineering, The Maharaja Sayajirao University of Baroda, Vadodara, India. (E-mail: shahtejal11@gmail.com)

[2]Department of Applied Mathematics, Faculty of Technology and Engineering, The Maharaja Sayajirao University of Baroda, Vadodara, India. (E-mail: jaita.sharma-appmath@msubaroda.ac.in)

Article History:

Received: *27-10-2024*

Revised:*26-11-2024*

Accepted:*28-12-2024*

Abstract: A topic of interest in financial mathematics is the Black-Scholes model. However, the underlying asset price in the stock market may not be satisfied by this linear model, which was developed under a number of assumptions, including liquidity and the absence of transaction costs. The linear model has restricted its precision in actual market conditions. We study the transaction cost model for modelling illiquid markets from the extended nonlinear Black-Scholes model. Using a semi-discretization finite difference approach, the nonlinear partial differential equation is transformed into a nonlinear ordinary differential equation. Deep Learning (DL) is an advanced technique of machine learning solves the converted ordinary differential equation by fully connected neural network (FCNN). By modelling the complex and nonlinear relationships among market variables, DL models can generate option pricing forecasts that are more dependable and precise, not just for continuous data but also for discontinuous data (at jump point).

Introduction: Nonlinear partial differential equation plays a crucial role in financial modelling, especially in the pricing of derivatives like options. The Black- Scholes model, introduced in 1973, continues to be one of the most widely used frameworks for pricing European options. However, this model is a linear one and offers an analytical solution, yet it is not appropriate for the complexities of real market assumptions that exhibit nonlinear effects. We examine the transaction cost model for modelling illiquid markets from the extended nonlinear Black-Scholes model created by Seelama et al. (2021). First using a semi-discretization finite difference approach, the nonlinear partial differential equation is transformed into a nonlinear ordinary differential equation. Solves the converted ordinary differential equation by Deep Learning (DL) based fully connected neural network (FCNN) algorithm. This algorithm is capable of handling the nonlinear behaviour of model and produce more accurate option value for European call.

Objectives: Find the solution of more realistic nonlinear model of Black-Scholes equation include transaction cost in illiquid market with deep learning algorithm for a European call option.

Methods: From extended nonlinear Black-Scholes model, the nonlinear model of transaction cost in illiquid market is considered for study. The nonlinear partial differential equation is converted into a nonlinear ordinary differential equation by semi-discretization finite difference method. DL is a sophisticated machine learning technique that solves transformed ordinary differential equations using fully connected neural network (FCNN) algorithm. DL algorithm uses a Python program.

Results: For European call option, option values are predicted for different number of neurons with different loss functions like MSE and MAE at the time of maturity. Graphical representation shows the accuracy of the algorithm at continuous as well as at jump point (strike price).

Conclusions: The method of solving a complex nonlinear partial differential equation by transforming it into a nonlinear ordinary differential equation is valuable. More precise pricing estimates for option values with nonlinear effects in financial data could be improved by deep learning.

Keywords: Nonlinear Black-Scholes equation, transaction costs, illiquid markets, deep learning.

1. Introduction

Derivatives are financial tools that provide the option to purchase or sell an underlying asset at a later date. These instruments, including options, futures, swaps and forwards, were utilized for speculation and risk management in an investment. An option is a financial agreement that provides option holders the privilege to purchase or sell an underlying from option writers at a predetermined date and price. The agreement that grants option holders the ability to purchase an underlying asset is referred to as a call option, whereas the agreement that allows option holders to sell an underlying asset as a put option.

In 1973, Fischer Black and Myron Scholes [1] constructed the Black-Scholes model for determining prices of options. However, their model required various assumptions such as constant volatility, no transaction costs and perfect liquidity. But this model is not best suited for real financial market. In real world, options are generally illiquid. Also underlying assets prices change randomly with jump. Therefore, many researchers tried to develop more realistic model of Black-Scholes by changing some assumptions. Transaction cost is also considered in actual market. Volatility affects the option prices and its knowledge can help buffer against losses. The Black-Scholes model is updated by considering transaction cost and volatility (see,[2],[3],[4],[5],[6],[7]). Illiquid describes the condition of a stock, bond, or other assets that cannot be quickly or easily sold or converted to cash without incurring a significant loss in value. Illiquid assets can be challenging to sell promptly due to minimal trading activity or interest in the matter, signified by an absence of eager and willing investors or speculators looking to buy or sell the asset. Consequently, illiquid assets usually exhibit reduced trading volume, broader bid-ask spreads, and heightened price volatility. In 2005, generalised model of Black-Scholes in illiquid market was derived. ([8]). Presence of price impact has been also studied by researchers ([9],[10]). In 2013, model ([8]) was revised and add illiquidity with jump ([11]). In 2016, illiquid market with transaction cost model was derived ([12]). In 2021, the idea of ([11] and [12]) was combined and Black-Scholes model with transaction cost with jumps in illiquid market was derived ([13]). The model derived in ([13]) is nonlinear and more realistic to the real financial market.

Differential equations have been solved using numerical methods. Optimization methods like least squares finite element methods [14],[15]), and element free Galerkin methods [16] have been used. These methods are based on mesh-free formulations. Theoretical convergence criteria for both the methods have been examined [17]. These concepts were applied to neural networks in [18], though using neural networks in this context has recently experienced a resurgence of interest as seen in ([18],[19],[20],[21],[22],[23]). These recent works have shown that remarkably simple implementations of deep neural networks can be used to solve relatively broad classes of differential equations.

In this paper, from the extended model of ([13]), transaction cost model with illiquidity is used. According to ([24]), nonlinear model was converted into nonlinear ordinary differential equation using semi discretization technique. The converted nonlinear ordinary differential equation was solved using the fourth order Runge-Kutta- Fehlberg integration technique. In this method both the variables transaction costs and liquidity parameter were taken in a range from 0 to 0.03. One variable keeping fix and other is vary in a given range. So eventually both are treated as constants. In this paper both transaction costs and liquidity parameter are considered as variables. Transaction costs is defined in a function form and liquidity parameter randomly vary in a range from 0 to 0.03. So, the proposed model is completely nonlinear. The converted nonlinear ordinary differential equation is solved using deep learning based fully connected neural network. Experimental results are used to check accuracy of the proposed method.

2. Objectives

Two types of assets are generally traded in a real financial market. One is a risk-free asset and second is risky asset. Let A_t denote the risk-free asset price and S_t denote the risky asset prices at time t where t > 0. Let T be the time of maturity, K be the striking price and $h(S_T) = max\{S_T - K, 0\}$ be the pay-off at time T that is at time of maturity of the option.

In 1973, Fischer Black and Myron Scholes [1] derived the linear Black-Scholes model for option pricing. According to this model the risk-free asset price A_t follows

$$dA_t = rA_t dt \tag{1}$$

where r is the risk-free interest rate. Also, the price of risky asset S_t satisfies

$$dS_t = S_t(\mu dt + \sigma dW_t) \tag{2}$$

where μ is the constant drift and σ is the constant volatility, W_t is a standard one-dimensional Brownian motion. According to ([13]), for transaction cost with jumps in illiquid market, the price of the risky asset is generated by the following stochastic differential equation:

$$dS_t = S_t(\mu(t, S_t)dt + \sigma(t, S_t)(dW_t + adM_t) + \lambda(t, S_t)d\theta_t + k(t, St)d\theta_t) \tag{3}$$

and θ_t satisfies

$$d\theta_t = \eta_t dt + \zeta_t(dW_t + bdM_t) \tag{4}$$

where $r(t, S_t)$ is the interest rate, $\mu(t, S_t)$ is the drift, $\sigma(t, S_t)$ is the volatility, a and b are real constants, $k(t, S_t)$ is the transaction costs, $M_t = N_t - \rho t$ is the compensated Poisson process where N_t is a Poisson process with deterministic intensity ρ, η_t and ζ_t are adapted process to a filtration generated by the Brownian motion, $\lambda(t, S_t)$ is price impact function of the trader (non-negative) and θ_t is the number of shares.

Theorem: The nonlinear partial differential equation of Black-Scholes with transaction costs in illiquid market with jumps for the European call option price $C(t, S_t)$ at time $t \in [0, T]$ and stock value S_t satisfies the Equation (3) and (4) is given by

$$\begin{aligned} & r(t,S_t)V_t + \theta_t S_t[\mu(t,S_t) - r(t,S_t) + \lambda(t,S_t)\eta_t + k(t,S_t)\eta_t] \\ & = \partial_t C(t,S_t) + (\mu(t,S_t) + \lambda(t,S_t)\eta_t + k(t,S_t)\eta_t \\ & \qquad - \rho[a\sigma(t,S_t) + b\lambda(t,S_t)\zeta_t + bk(t,S_t)\zeta_t])S_t\partial_s C(t,S_t) \\ & \quad + \frac{1}{2}(\sigma(t,S_t) + \lambda(t,S_t)\zeta_t + k(t,S_t)\zeta_t)^2 S_t^2 \partial_{ss}^2 C(t,S_t) + \\ & \rho\left(C\left(t,S_t^-(1 + a\sigma(t,S_t) + b\lambda(t,S_t)\zeta_t + bk(t,S_t)\zeta_t)\right) - C(t,S_t^-)\right) \end{aligned} \tag{5}$$

with the terminal condition $C(T,S_T) = h(S_T)$.

In the above equation put a=b=0, i.e. jump is cancelled. We get nonlinear transaction cost model (partial differential equation) in illiquid market mentioned below.

$$\frac{\partial c}{\partial t} + \frac{\sigma^2 S^2}{2\left[1 - \lambda(t,S_t) + k(t,S_t)S\frac{\partial^2 c}{\partial S^2}\right]^2} \frac{\partial^2 c}{\partial S^2} + rS\frac{\partial c}{\partial s} - rC = 0 \tag{6}$$

with the terminal and boundary conditions for European call options

$$C\big(T,S(T)\big) = max\left(S(t) - K, 0\right)$$

$$C(t,L) = L - Ke^{-r(T-t)}, \big(\forall L > S(T)\big)$$

$$C(t,0) = 0$$

The objective of the paper is to solve the above nonlinear partial differential equation using soft computing technique like deep learning based fully connected neural network (FCNN) with good accuracy.

3. Methods

To find the solution of the above nonlinear model, the procedure is divided in two parts.

(1) Convert the nonlinear partial differential equation of Black-Scholes with transaction costs in illiquid market into nonlinear ordinary differential equation: First the nonlinear partial differential equation is converted into nonlinear ordinary differential equation with semi discretization finite difference technique ([25]). This method is also known as the method of lines. Discretize S in the interval $[0, S_{max}]$ into N equal parts with grid size $\Delta S = \frac{S_{max}}{N}$. First spatial derivative and second spatial derivative in equation (6) are approximated by central finite differences with second order. Let $Ci(t,Si)$ be the approximation of option value. Also take $\lambda_i\big(t,S(t)\big) = \vartheta$ the liquidity parameter and $K_i\big(t,S(t)\big) = \frac{S_i q^2 e^{r(T-t)}}{2}$ is the transaction costs where q is the proportional transaction cost ([26]).

$$\frac{dC_i}{dt} + \frac{\sigma^2 S_i^{\,2}}{2\left[1 - \{\lambda(t,S_t) + k(t,S_t)\}S_i\}\left(\frac{C_{i+1} - 2C_i + C_{i-1}}{(\Delta S)^2}\right)\right]^2}\left(\frac{C_{i+1} - 2C_i + C_{i-1}}{(\Delta S)^2}\right) + rS_i\left(\frac{C_{i+1} - C_{i-1}}{2\Delta S}\right) - rC_i = 0 \tag{7}$$

With the terminal and boundary conditions for the European call option given as

$$C\big(T,S(T)\big) = max\left(S_i(t) - K, 0\right)$$

$$C(t,L) = S_{max} - Ke^{-r(T-t)}, \big(\forall L > S(T)\big) \tag{8}$$

$C(t,0) = 0$

(2) Converted nonlinear ordinary differential equation is solved using the deep learning-based algorithm of fully connected neural network (FCNN): Equation (7) with given conditions in equation (8) is now solved using FCNN with deep learning. FCNN refers to fully connected neural network (FCNN), a specific architecture within Deep Learning. It is among the easiest and most frequently utilized forms of neural networks, usually used for purposes such as regression, classification, and representation learning.

2.1 Features of FCNN:

Universal Approximation: FCNNs can approximate any continuous function, with enough neurons and layers. This characteristic makes them strong for a wide variety of tasks where complex relationships exist between inputs and outputs.

FCNN features fully connected layers, where every neuron in one layer connects to all neurons in the subsequent layers. This makes it "fully connected". In TensorFlow/Keras dense layers reflects these layers. Typically, each layer incorporates an activation such as ReLU, sigmoid or tanh that introduces nonlinearity, following the network to understand complex patterns. FCNN lacks any convolutional or Pooling layers. The input data is considered a single flat vector rather than a spatial configuration (such as image data). The information moves through multiple fully connected layers, where each layer applies transformations as well as activation functions to the input.

2.2 Mathematical Steps in FCNN:

Input vectors can be represented as 1-dimensional vector. Here $S = (s_1, s_2, \ldots, s_n)$ and $t = (t_1, t_2, \ldots, t_n)$. The fundamental mathematical procedure in FCNN involves matrix multiplication, succeeded by activation functions in hidden layers. Compute the intermediate representation

$$H(i) = \sigma(i)[W(i)S + b(i)]$$

$H(i)$ is the output of i^{th} hidden layer, $\sigma(i)$ = activation function of i^{th} layer, $W(i)$ = weight matrix of i^{th} layer and $b(i)$ is a bias vector of i^{th} layer. Compute the output

$$O = H(i) = \sigma(i)[W(i)H(i-1) + b(i)]$$

Now concept of Backpropagation is used. First loss is calculated based on true value and predicted value. Then gradient of loss function with respect to W is calculated. Then algorithm modifies weight W and biases b by applying gradients to reduce the loss function L. A loss function is also known as cost function is a mathematical function that quantifies the cost or error associated with a set of data points. $W \leftarrow W - \eta \frac{\partial L}{\partial W}$. Here η is a learning rate and $\frac{\partial L}{\partial W}$ is a gradient of the loss function with respect to W. When utilizing FCNNs, it may be necessary to adjust Hyperparameter as well. Number of neurons in each layer are adjusted. Adding more layers and neurons enhances capacity but may lead to overfitting. Learning rate is also adjusted as it affects how quickly weights are adjusted and algorithm converge fast. Regularization governs the punishment imposed on substantial weights to avoid overfitting. L_2 regularization (**Ridge regularization**) is applied here.

$loss = original\ loss\ +\ \lambda \left|\left|\theta(W,b)\right|\right|^2$, where λ regulates the significance of regularization.

Data is imported in batch. Incorporate batch normalization to enhance training stability and accelerate the process. Generally adaptive optimizers like Adam and RMSprop for better convergence are used in FCNN.

4. Results

The mentioned above nonlinear model is computed for risk-free interest rate r = 0.05, volatility σ = 0.3, strike price K = 40, S_{max} = 70, proportional transaction cost q = 0.01 and time of maturity T =1 year. ϑ illiquid parameter takes the value in a range of 0 to 0.03. For hyperparameter setting, in each layer different number of neurons are taken for number of hidden layers = 6. L_2 regularization with learning rate $\eta = 0.001$ is applied. Adam optimizer is used with batch size 16. Exponential Linear Unit (ELU) activation function is used in hidden layers and linear function as an activation is used in output layer. Here, loss function is defined with a custom term that is transaction costs. Mathematically, $loss = (C_{true} - C_{predicted})^2$ + transaction costs

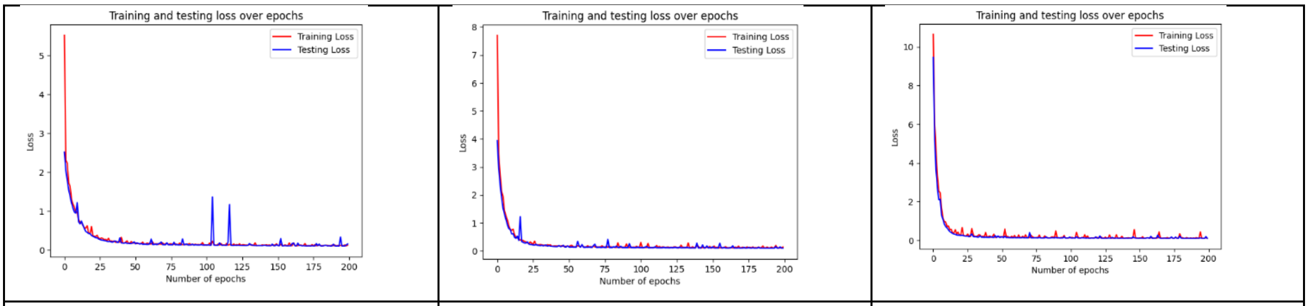
Table 1 illustrates for loss function defined by Mean Squared Error (MSE). It shows that as the number of neurons changes the accuracy changes. It shows best predicted call option value for 200 number of neurons.

Table:1

S	**C_Target**	**No. of neurons**		
		64	**100**	**200**
		C_Predicted		
40.6	0.6	0.79160	0.70215	0.76630
49	9.0	8.98727	8.93354	9.00123
56	16.0	15.99862	15.97565	16.04225
63	23.0	22.98522	22.99294	23.04190
67.2	27.2	27.15376	27.17203	27.22907

Figure 1 represents the graphs of number of neurons in hidden layers verses loss. It shows the best result for 200 number of neurons.

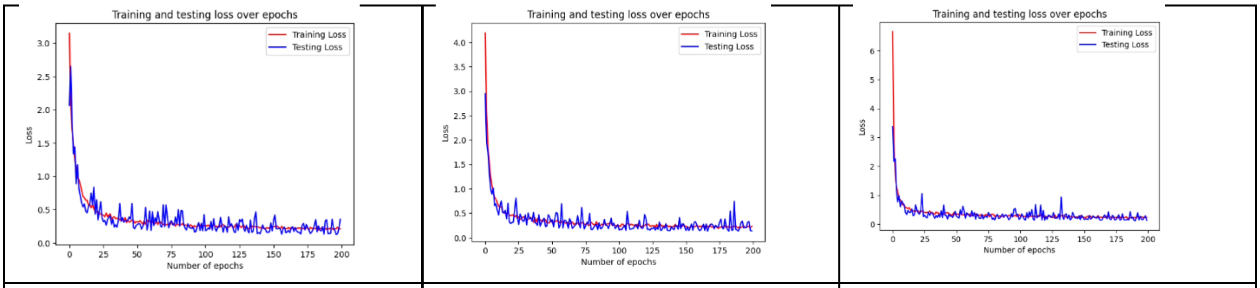
Figure 1



(a)No. of neurons = 64 | **(b) No. of neurons = 100** | **(c) No. of neurons = 200**

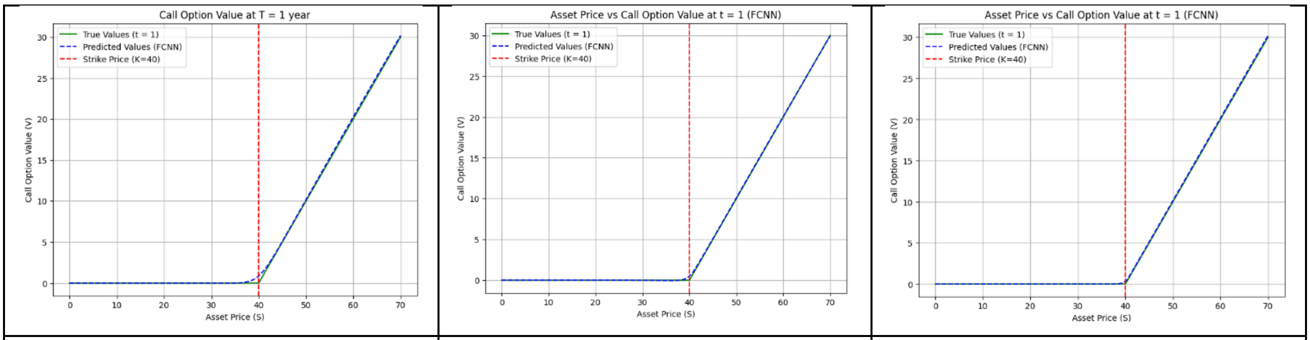
Figure 2 shows the behaviour of loss function evaluated by Mean Absolute Error (MAE).

Figure 2



(a)No. of neurons = 64 | **(b) No. of neurons = 100** | **(c) No. of neurons = 200**

Figure 1 and Figure 2 indicate that the loss function computed with MSE outperforms the one calculated with MAE. Both loss functions show best result for 200 number of neurons in hidden layers, but MSE achieves best more swiftly than MAE. Figure 3 presents the effect of transaction costs and illiquid market position (liquidity parameter) for European call option with loss MSE. It indicates that as the transaction costs and liquidity parameter increase value of the call option also increases. It also illustrates that by changing the number of neurons in hidden layers, accuracy at jump point that is at strike price K=40 can also increase.



(a)No. of neurons = 64 | **(b) No. of neurons = 100** | **(c) No. of neurons = 200**

Figure 3: European call option values verses asset price S for strike price K = 40, volatility σ = 0.3, risk-free interest rate r = 0.05, S_max = 70, time to maturity T = 1 year.

Also, for European call options, the spatio-temporal dynamics is presented for mentioned nonlinear effects of financial market in Figure 4. It is noted that a rise in the price of transaction costs and liquidity parameter the underlying asset elevates the value of the European call option.

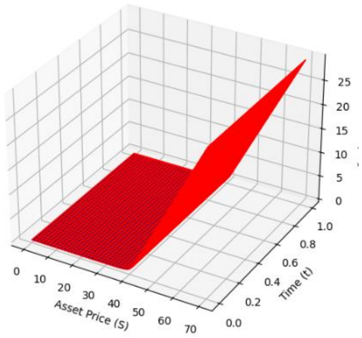


Figure 4 European call option value

5. Discussion

In this paper, we propose a method to find solution of nonlinear model of Black-Scholes with transaction costs and illiquid market for European call option with given terminal boundary conditions. The proposed method shows how the nonlinear partial differential equation of Black-Scholes is converted into nonlinear ordinary differential equation using semidiscretization technique of finite difference method. We have used Deep Learning based fully connected neural network (FCNN) to solve converted nonlinear ordinary differential equation model of option pricing. The use of the FCNN presents a unique method for pricing option. Experimental results verify that the suggested method can effectively address the Black-Scholes model for forecasting European call options. Also at jump condition, this method provides best accuracy in option price. In conclusion, we can assert that the suggested method enables option holder to effectively hedge the risk in financial market, considering nonlinear factors such as transaction costs and illiquid markets in the case of European call options and will gain significantly because the call options are more probable to finish in the money.

References

[1] Black, F. and Scholes, M. (1973) The Pricing of Options and Corporate Liabilities. Journal of Political Economy, 81, 637-654. https://doi.org/10.1086/260062

[2] Dokuchaev, N.G. and Savkin, A.V. (1998) The Pricing of Options in a Financial Market Model with Transaction Costs and Uncertain Volatility. Journal of Multinational Financial Management, 8, 353-364. https://doi.org/10.1016/S1042-444X(98)00036-X

[3] Florescu, I., Mariant, M.C. and Sengupta, I. (2014) Option Pricing with Transaction Costs and Stochastic Volatility. Electronic Journal of Differential Equations, 2014, 1-19

[4] K. Ronnie Sircar & George Papanicolaou. (1998) General Black-Scholes models accounting for increased market volatility from hedging strategies, Applied Mathematical Finance, 5:1, 45-82, DOI: 10.1080/135048698334727

[5] Gulen, Seda and Popescu, Catalin and Sari, Murat. (2019) A new approach for the black-scholes model with linear and nonlinear volatilities, Journal of Mathematics,7(8),760.

[6] Davis, Mark HA and Panas, Vassilios G and Zariphopoulou, Thaleia. (1993) European option pricing with transaction costs, SIAM Journal on Control and Optimization,31(2), 470-493.

[7] Barles, Guy and Soner, Halil Mete. (1998) Option pricing with transaction costs and a nonlinear Black-Scholes equation, Finance and Stochastics,2, 369-397.

[8] Liu, H. and Yong, J. (2005) Option Pricing with an Illiquid Asset Market. Journal of Economic Dynamics & Control, 29, 2125-2156. https://doi.org/10.1016/j.jedc.2004.11.004

[9] Glover, K.J., Duck, P.W. and Newton, D.P. (2010) On Nonlinear Models of Markets with Finite Liquidity: Some Cautionary Notes. SIAM Journal on Applied Mathematics, 70, 3252-3271. https://doi.org/10.1137/080736119

[10] Pirvu, T.A. and Yazdanian, A. (2015) Numerical Analysis for Spread Option Pricing Model in Illiquid underlying Asset Market: Full Feedback Model. Applied Mathematics & Information Sciences, 1271-1281. https://doi.org/10.18576/amis/100406

[11] El-Khatib, Y. and Hatemi-J, A. (2013) On Option Pricing in Illiquid Markets with Jumps. International Scholarly Research Notices, 2013, Article ID 567071. https://doi.org/10.1155/2013/567071

[12] Agana, F., Makinde, O.D. and Theuri, D.M. (2016) Numerical Treatment of a Generalized Black-Scholes Model for Options Pricing in an Illiquid Financial Market with Transections Costs. Global Journal of Pure and Applied Mathematics, 12, 4349-4361.

[13] Praewnapa Seelama, Dawud Thongtha. (2021) Option Pricing Model with Transaction Costs and Jumps in Illiquid Markets. Journal of Mathematical Finance, 11, 361-372.

[14] Strang, G., Fix, G. J., 1973. An analysis of the finite element method. Prentice-Hall, Inc., Englewood Cliffs, N. J., prentice-Hall Series in Automatic Computation.

[15] Bochev, P. B., Gunzburger, M. D. (2009) Least-squares finite element methods. Vol. 166 of Applied Mathematical Sciences. Springer, New York. https://doi-org.ezproxy.lib.utexas.edu/10.1007

[16] Li, X. (2016) Error estimates for the moving least-square approximation and the element-free Galerkin method in n-dimensional spaces. Applied Numerical Mathematics. An IMACS Journal 99, 77–97. https://doi-org.ezproxy.lib.utexas.edu/10.1016

[17] Babuˇska, I., Banerjee, U., Osborn, J. E. (2003) Survey of meshless and generalized finite element methods: a unified approach. Acta Numerica 12, 1–125. https://doi-org.ezproxy.lib.utexas.edu/10.1017/S0962492902000090

[18] Lagaris, I. E., Likas, A., Fotiadis, D. I. (1998) Artificial neural networks for solving ordinary and partial differential equations. IEEE Transactions on Neural Networks 9 (5), 987–1000.

[19] Han, J., Jentzen, A., E, W. (2018) Solving high-dimensional partial differential equations using deep learning. Proceedings of the National Academy of Sciences 115 (34), 8505–8510. https://www.pnas.org/content/115/34/8505

[20] Berg, J., NystrA˝um, K. (2019) Data-driven discovery of pdes in complex datasets. Journal of Computational Physics 384, 239 – 252. http://www.sciencedirect.com/science/article/pii/S0021999119300944

[21] Nabian, M. A., Meidani, H. (2019) Physics-Driven Regularization of Deep Neural Networks for Enhanced Engineering Design and Analysis. Journal of Computing and Information Science in Engineering 20 (1), 011006. URL https://doi.org/10.1115/1.4044507

[22] Raissi, M., Perdikaris, P., Karniadakis, G. (2019) Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics 378, 686 – 707. http://www.sciencedirect.com/science/article/pii

[23] Sirignano, J., Spiliopoulos, K.(2018) DGM: A deep learning algorithm for solving partial differential equations. Journal of Computational Physics 375, 1339 – 1364. http://www.sciencedirect.com/science/article/pii/S0021999118305527

[24] Agana, F., Makinde, O.D. and Theuri, D.M. (2016) Numerical Treatment of a Generalized Black-Scholes Model for Options Pricing in an Illiquid Financial Market with Transections Costs. Global Journal of Pure and Applied Mathematics, 12, 4349-4361.

[25] R. LBurden and J.D. Faires. (2010) Numerical Analysis. Cengage Learning.

[26] Barles, Guy, and Halil Mete Soner. "Option pricing with transaction costs and a nonlinear Black-Scholes equation." Finance and Stochastics 2 (1998): 369-397.