Research Paper
Computational Methods for Differential Equations
http://cmde.tabrizu.ac.ir
Vol. *, No. *, *, pp. 1-21
DOI:10.22034/cmde.2026.71605.3598

Modified cubic B-spline based differential quadrature methods for the time-fractional Black-Scholes equation

Nizamudheen V[1], Riyasudheen TK[2,*], Noufal Asharaf[3], and Shefeeq T[1]

[1]Department of Mathematics, Farook College (Autonomous) affiliated to University of Calicut, Kozhikode, 673632, India.

[2]Department of Computational Science and Humanities, Indian Institute of Information Technology Kottayam, Valavoor, Kottayam 686635, Kerala, India.

[3]Department of Mathematics, CUSAT (Cochin University of Science and Technology), Cochin, 682022, India.

Abstract

The time-fractional Black–Scholes equation (TFBSE) is intended to price the options for which the underlying price fluctuates within a correlated fractal transmission system. Although the TFBSE is an influential approach for grasping the long-term memory traits of financial markets, the non-local nature of fractional derivatives makes significant challenges in finding an accurate solution. We perform an efficient use of the differential quadrature method (DQM) based on modified cubic B-splines to solve the TFBSE governing European options. This paper constructs an algorithm by the combination of time fractional discretization using the finite difference method $L1$ and space discretization using the modified cubic B-spline-based differential quadrature method. Uniform meshes are considered for the discretization of both temporal and spatial domains. Theoretical stability has been established by finding an estimate for the maximum norm of the inverse operator regardless of the involvement of the mesh parameters. We trigger the Nuemann series theorem to obtain a uniform bound for the inverse operator under reasonable conditions on the mesh parameters. The numerical illustrations show that this implicit numerical method exhibits a fourth-order convergence in the space direction and the order $2-\alpha$ in time. Moreover, we observe an enhancement in order of spatial convergence whenever α tends to zero. The results obtained are then compared with existing popular techniques to demonstrate the accuracy of modified cubic B-spline-based DQM.

Keywords. Time-fractional Black-Scholes model, Differential quadrature method, Modified cubic B-splines, Nuemann series theorem.
2010 Mathematics Subject Classification. 65M12, 26A33, 91G60.

1. Introduction

The rapidly expanding derivative markets have long attracted the research interest of financial mathematicians, policymakers, and market participants. It directly yields more notable contributions to the economic system as a whole.

Historically, financial derivatives and their specialized forms originated in ancient Greek philosophy. Options are financial derivatives structured as contracts between two parties, enabling the potential transaction of an asset at a predetermined 'strike price' before the 'expiration' date. In the current financial market scenario, option trading has enriched the derivative market, specifically in a virtual fashion. Quite futuristic hedging strategies are required to exercise the options effectively. Option valuation theories estimate the value of options by assigning a price, referred to as the 'premium'. Finding a fair value for options had been defiance for the financial world until the invention of the Black-Scholes formula [13]. Myron Scholes and Robert C Merton [41] have, in combined work with the late Fischer Black, deduced an instigating formula later to be known as the Black-Scholes-Merton formula for pricing stock options.

Received: 17 February 2025; Accepted: 06 June 2026.
∗ Corresponding author. Email: riyasudheen@iiitkottayam.ac.in.

In 1997, they were awarded the Nobel Prize in economics sciences for developing this conceptual framework called the Black–Scholes model (BSM). In [13], Black and Scholes demonstrated that a second-order parabolic partial differential equation with respect to time and stock price, known as the Black–Scholes equation (BSE), governs the value of a European option on a stock whose price follows a geometric Brownian motion with constant drift and volatility. Various numerical techniques such as Monte-Carlo simulations [14], lattice methods [15], finite difference methods (FDM) [6, 7, 30, 31], finite element methods (FEM) [25], finite volume methods (FVM) [55, 57] are exerted on the classical Black-Scholes equations and its various generalizations due to the inadequacy of analytical methodologies. Safdari-Vaighani et al. [50] proposed a radial basis function partition of unity method for option pricing under a two-state continuous CAPM model, demonstrating competitive accuracy compared with finite difference methods. S. Abdi-Mazraeh et al. [2] considered Lagrange's polynomial-based multiple shooting method to solve BSE in space direction. Additionally, to tackle the reduction in the convergence order of the approximate solution caused by the non-smoothness of the payoff at the strike price. In this study, a variable step-size strategy is employed for time discretization. An A-stable method for effectively solving the Black–Scholes equation (BSE) was proposed in [1]. The method combines a domain decomposition algorithm based on Chebyshev polynomials for spatial discretization with a predictor-corrector scheme employing a variable step-size strategy for time discretization.

However, the assumptions of the classical Black–Scholes model are often too restrictive to accurately reflect real financial markets, particularly in capturing sudden price jumps and complex asset dynamics over short time intervals. A major limitation of the classical framework is its reliance on integer-order derivatives, which describe only local behavior [44]. In contrast, fractional calculus provides a natural framework for modeling memory and nonlocal effects, where the current state depends not only on present conditions but also on past states. Fractional models have gained significant attention for their ability to capture long-term memory dependence, rough volatility, self-similarity, and non-Markovian behavior in financial systems. Consequently, fractional Brownian motion and fractional derivative operators, together with the Hurst parameter [33], have been incorporated into option pricing models to account for memory effects in financial markets. The significant contributions by Wyss [58] increased the popularity of fractional BS models. Rather than this model, several fractional Black-Scholes-type models have been developed, including FMLS [20], KOBol [18] and Jumarie's [19] formulations, to describe long-memory characteristics in asset price dynamics. Furthermore, empirical studies [16, 35, 40] indicate the presence of jumps and stochastic volatility in financial markets, motivating the development of more realistic fractional extensions involving stochastic volatility and jump-diffusion processes. In this direction, Ahmadian et al. [4] proposed an efficient numerical method for pricing options in the Black–Scholes model with jumps using a finite difference framework combined with fixed-point iteration and repeated space-time Richardson extrapolation techniques.

To address the computational challenges arising in fractional and time-fractional financial models, various numerical techniques have been developed for solving fractional differential equations (FDEs). Fractional backward difference formulae (FBDF) are a numerical technique first used by Lubich [39] to solve FDE and later some researchers have tried its variations to improve the precision and the order of convergence. In [27] a new generating function is constructed based on fractional linear barycentric rational interpolation (FLBRI). This method is a modification of backward difference formulae (BDF) which improves the stability, accuracy, and convergence order compared with FBDF. Lubich et al. [39] introduced multistep fractional linear methods to solve fractional differential equations (FDEs) of order $0 < \alpha < 1$. S. Irandoust-Pakchin et al. [28] further developed second-order fractional linear multistep methods to solve FDEs of order $1 < \alpha < 2$. S. Irandoust-pakchin et al. [26] developed a method based on He's variational iteration method to solve FDEs. Beyond FDEs, stochastic differential equations (SDE) are also commonly used to model uncertainty in fields such as finance, economics, engineering, and applied sciences. In this context, A. Rathinasamy et al. [46] developed second-order balanced stochastic Runge–Kutta methods to provide stable and efficient numerical solutions for SDE.

With the broad application of fractional BS equations in option pricing, there has been growing interest in developing efficient solution techniques. Since the beginning of this century, numerous studies have investigated both analytical and numerical approaches for solving these equations. Chen et al. [17] derived an explicit closed-form analytical solution for pricing double barrier options using the eigenfunction expansion method combined with the Laplace transform. Jicheng Yu [60] applied Lie symmetry analysis to the TFBSE through the invariant subspace method. The successful

application of the homotopy analysis method (HAM) to the TFBSE was demonstrated in the work of S. E. Fadugba [22]. Asma et al. [21] compared two approaches for fractional models: one combining the homotopy perturbation method (HPM), the Sumudu transform, and Ji-Huan He polynomials, and the other based on the homotopy Laplace transform perturbation method. However, these analytical methods generally involve infinite series, integrals, or convolutions with certain special functions. Moreover, in many cases, obtaining an explicit analytical solution for the fractional BSE remains a challenging task. Therefore, various numerical methods have been developed in recent years. The methods are based on finite differences, finite elements, and the spectral approach. An RBF-based mesh-free method is developed by A. Golbabai et al. in combination with a finite difference method of order $\mathcal{O}(t^{2-\alpha})$ over time. Nuugulu et al. [43] proposed a first-order implicit FDM to solve the constructed TFBSE. Zhang [61] used another implicit finite difference method by discretizing the spatial derivative by central finite difference and the L1 scheme for the time derivative, having the spatial order of convergence 2. In [47] we obtained a numerical solution based on the implicit difference scheme with the price of the European option with transaction costs. The work in [36] introduces a weighted FDM for the subdiffusive Black-Scholes (B-S) model. The Crank-Nicolson mixed alternate segment (MASC-N) scheme is applied by Yang[59]. This scheme has a spatial order of convergence of 2. T Akram et al. [5] generalized the concept of the B-spline collocation method using extended cubic B-splines (ECBS). Zhaowei et al. [54] presented a compact quadratic spline collocation method for fractional option pricing models. Rather than these methods, TFBSEs are numerically solved using other various methods, including, not limited to, meshless methods [45], spline interpolation method [23, 48], and numerical techniques with exponential convergence [24].

In this work, we consider the time fractional BS model [17, 37] as follows:

$$\frac{\partial^\alpha Q(S,\tau)}{\partial \tau^\alpha} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 Q(S,\tau)}{\partial S^2} + rS\frac{\partial Q(S,\tau)}{\partial S} - rQ(S,\tau) = 0, \quad (S,\tau) = 0 \in (0,\infty)\times(0,T), \tag{1.1}$$

with initial and boundary conditions:

$$Q(0,\tau) = f(\tau), \quad Q(\infty,\tau) = g(\tau), \quad Q(S,T) = h(S),$$

where $0 < \alpha \leq 1$, $\sigma \geq 0$ denote the volatility of the returns derived from the holding stock price S, r is the risk-free rate and T is the expiry time. The fractional derivative operator in (1.1) is a modified right Riemann–Liouville derivative which is defined as:

$$\frac{\partial^\alpha Q(S,\tau)}{\partial \tau^\alpha} = \frac{1}{\Gamma(1-\alpha)}\frac{d}{d\tau}\int_\tau^T \frac{Q(S,\xi)-Q(S,T)}{(\xi-\tau)^\alpha}d\xi, \quad 0<\alpha<1. \tag{1.2}$$

For $\alpha = 1$, we get the classical BS model. Let $\eta = T - \tau$, and $\xi = T - \eta$, then for $0 < \alpha < 1$, rewriting (1.2), we have

$$\frac{\partial^\alpha Q(S,\tau)}{\partial \tau^\alpha} = \frac{-1}{\Gamma(1-\alpha)}\frac{d}{d\eta}\int_0^\eta \frac{Q(S,T-\zeta)-Q(S,T)}{(\eta-\zeta)^\alpha}d\zeta.$$

Moreover, defining

$$s = \ln S \text{ and } u(s,\eta) = Q(e^s, T-\eta),$$

the model (1.1) can be expressed as

$${}_0D_\eta^\alpha u(s,\eta) = \frac{1}{2}\sigma^2\frac{\partial^2 u(s,\eta)}{\partial s^2} + \left(r - \frac{1}{2}\sigma^2\right)\frac{\partial u(s,\eta)}{\partial s} - ru(s,\eta), \tag{1.3}$$

with boundary conditions: $u(-\infty,\eta) = f(\eta)$, $u(\infty,\eta) = g(\eta)$, $u(s,0) = h(s)$, where the fractional derivative is:

$${}_0D_\eta^\alpha u(s,\eta) = \frac{1}{\Gamma(1-\alpha)}\frac{d}{d\eta}\int_0^\eta \frac{u(s,\eta)-u(s,0)}{(\eta-\zeta)^\alpha}d\zeta, \quad (0<\alpha<1). \tag{1.4}$$

In order to solve the above model numerically, it is necessary to truncate the original unbounded domain into a finite interval. For this, we restrict the range of variable 's' in problem (1.1) to a finite interval (B_x, B_y); then it will be of

the following form:

$$ {}_0D_\eta^\alpha u(s,\eta) = a\frac{\partial^2 u(s,\eta)}{\partial s^2} + b\frac{\partial u(s,\eta)}{\partial s} - cu(s,\eta) + f(s,\eta), \tag{1.5} $$

with boundary conditions: $u(B_x,\eta) = f(\eta),\ u(B_y,\eta) = g(\eta),\ u(s,0) = h(s)$, where

$$ a = \frac{1}{2}\sigma^2 > 0,\ b = r - a,\ c = r > 0. $$

Note that the source term $f(s,\eta)$ is added for numerical validation in (1.5). This is an advection-diffusion-reaction model in general, but it is well known that when $a > 0,\ b < 0,\ c = 0$, it is a time fractional advection-diffusion model. Also, if $a > 0,\ b = 0,\ c \neq 0$, then it is a reaction-diffusion model. In this context of TFBSE, we employ DQM based on modified cubic B-splines in conjunction with the very convenient time-fractional discretization method L1 [38]. In essence, DQM introduced by Bellman et al. [12], approximates the derivatives by expressing them as a weighted sum of the function values at selected node points within the problem domain. In DQM, various types of polynomials are used to obtain weights, including B-spline functions [32, 34], cubic and modified B-splines [8–10, 32, 42], quintic B-splines [9], Lagrange interpolation polynomials, Legendre polynomials, and Sinc functions, along with other related functions. Ali Bashan et al.[11] used the combination of the Crank-Nicolson scheme and the quintic B-spline-based DQM to find the numerical solution of a coupled KdV equation. In [9], DQM based on a modified cubic B-spline was implemented to obtain the numerical solutions for the nonlinear Schrödinger equation (NLS). For solving fractional stochastic integro-differential equations, operational matrices is constructed based on linear cardinal B-spline functions in [3]. Jiwari et al. [29] used the combination of Lagrange interpolation method and a DQM based on a modified set of cubic B-splines to find the numerical approximation of hyperbolic partial differential equations. This set of modified splines fails to preserve the optimal polynomial reproduction property near the frontier. Modified B-splines are locally adjusted versions of standard B-splines designed to maintain the spline's smoothness and accuracy while improving numerical properties; particularly near the boundary. A. Babu et al. [8] proposed a new modification of standard cubic splines that retains the optimal accuracy, higher order polynomial reproduction that is essential while approximating the solution of the Black-Scholes equation. This motivates us to study the BSE numerically using the modified set of cubic B-splines introduced by A. Babu [8], that has better reproduction property throughout the domain.

The rest of the content is organized as follows. Section 2 explains the discretization technique and develops an implicit numerical scheme. The theoretical stability is established in Section 3. In Section 4, we evaluate the proposed method using numerical examples. Section 5 concludes the article with highlights.

2. Numerical Scheme

This section outlines the numerical scheme, beginning with discretizations of the temporal and spatial derivatives. Following the discretization procedure, the resulting implicit scheme is formulated in matrix form. Subsequent subsections detail the computation of the weights employed in the Differential Quadrature Method (DQM).

2.1. **Discretization.**

Initially, we discretize the time-fractional derivative. For this partition the domain $[0,T]$ uniformly into N subintervals, $0 < \eta_0 < \eta_1 < \ldots < \eta_N = T$, where $\Delta\eta = k = \eta_n - \eta_{n-1} = \frac{T}{N}$ for $n = 1,2,3,\ldots,N$. Suppose $u(s,\eta) \in C^{(1)}$, for $0 < \alpha \leq 1$, the modified Riemann–Liouville derivative

$$ \begin{aligned} {}_0D_\eta^\alpha u(s,\eta) &= \frac{1}{\Gamma(1-\alpha)}\frac{d}{d\eta}\int_0^\eta \frac{u(s,\zeta)-u(s,0)}{(\eta-\zeta)^\alpha}d\zeta \\ &= \frac{1}{\Gamma(1-\alpha)}\frac{d}{d\eta}\left(\int_0^\eta \frac{u(s,\zeta)}{(\eta-\zeta)^\alpha}d\zeta - \int_0^\eta \frac{u(s,0)}{(\eta-\zeta)^\alpha}d\zeta\right) \\ &= \frac{1}{\Gamma(1-\alpha)}\int_0^\eta \frac{\partial u(s,\zeta)}{\partial\zeta}(\eta-\zeta)^{-\alpha}d\zeta \\ &= {}_0^C D_\eta^\alpha u(s,\eta). \end{aligned} $$

Here, the operator ${}_0^C D_\eta^\alpha u(s,\eta)$ is the Caputo derivative. Then, using the L1 discretization formula for the Caputo fractional derivative, the operator ${}_0D_\eta^\alpha u(s,\eta)$ at the point (s_i,η_{k+1}) can be approximated as:

$$\begin{aligned}
{}_0D_\eta^\alpha u(s_i,\eta_{k+1}) &= \frac{1}{\Gamma(1-\alpha)}\int_0^{\eta_{k+1}} \frac{\partial u\left(s_i,\zeta\right)}{\partial\zeta}\left(\eta_{k+1}-\zeta\right)^{-\alpha}d\zeta \\
&= \frac{1}{\Gamma(1-\alpha)}\sum_{p=0}^{k}\int_{(p-1)k}^{pk}\left(\frac{u_i^p-u_i^{p-1}}{k}+\mathcal{O}(k)\right)\left((n+1)k-\zeta\right)^{-\alpha}d\zeta \\
&= \frac{1}{\Gamma(1-\alpha)}\sum_{p=0}^{k}\left(\frac{u_i^p-u_i^{p-1}}{k}+\mathcal{O}(k)\right)\times\left(\frac{((n+1)k-pk)^{1-\alpha}-((n+1)k-(p+1)k)^{1-\alpha}}{1-\alpha}\right) \\
{}_0D_\eta^\alpha u(s_i,\eta_{k+1}) &= \frac{1}{\Gamma(2-\alpha)}\sum_{p=0}^{k}\left(\frac{u_i^p-u_i^{p-1}}{k}+\mathcal{O}(k)\right)\times\left(((n+1)k-pk)^{1-\alpha}-((n+1)k-(p+1)k)^{1-\alpha}\right),
\end{aligned}$$

in further simplification, we get the final L1 discretized formula as follows

$${}_0D_\eta^\alpha u(s_i,\eta_{k+1}) = \frac{\Delta\eta^{-\alpha}}{\Gamma(2-\alpha)}\sum_{p=0}^{k}\delta_p\left(u_i^{k+1-p}-u_i^{k-p}\right)+\mathcal{O}(\Delta\eta^{2-\alpha}), \tag{2.1}$$

where, $u_i^k=u(s_i,\eta_k)$, $\delta_p=(p+1)^{1-\alpha}-p^{1-\alpha}$. Moreover, the coefficients δ_p satisfy the following relations

i. $\delta_0=1$,
ii. $\delta_p>0$, for every $0\le p\le N$,
iii. $\delta_{p-1}>\delta_p$, for every $1\le p\le N$.

We discretize the spatial derivatives using the Differential Quadrature Method (DQM). Let $M\in\mathbb{Z}^+$, partition the spatial part of the domain into M subintervals. Let $a:=s_0<s_1<\ldots<s_M=:b$ be the corresponding $M+1$ node points with uniform step size $h=\dfrac{b-a}{M}$. Using the differential quadrature approximation, at all nodes s_i, $i=0,1,\ldots,M$, to the spatial derivatives, the derivatives take the form

$$u_s(s_i,\eta)=\sum_{j=0}^{M}p_{ij}^{(1)}u\left(s_j,\eta\right), \tag{2.2}$$

$$u_{ss}(s_i,\eta)=\sum_{j=0}^{M}p_{ij}^{(2)}u\left(s_j,\eta\right), \tag{2.3}$$

where, the coefficients $p_{ij}^{(1)},p_{ij}^{(2)}$, $i,j=0,1,\ldots,M$ are the weights for the approximation. The modified cubic B-splines constructed by A. Babu [8] are used to determine the weights. This methodology is discussed in the upcoming subsection. Combining the time discretization scheme (2.1) with spatial discretization (2.2) and (2.3) we can derive the full discretization scheme for (1.5) as follows:

$$\frac{\Delta\eta^{-\alpha}}{\Gamma(2-\alpha)}\sum_{p=0}^{k}\delta_p\left(u_i^{k+1-p}-u_i^{k-p}\right)=a\sum_{j=0}^{M}p_{ij}^{(2)}u_j^{k+1}+b\sum_{j=0}^{M}p_{ij}^{(1)}u_j^{k+1}+cu_i^{k+1}+f_i^{k+1}. \tag{2.4}$$

Multiplying by $d=\Delta\eta^\alpha\Gamma(2-\alpha)$ on both sides, we get

$$\sum_{p=0}^{k}\delta_p\left(u_i^{k+1-p}-u_i^{k-p}\right)=ad\sum_{j=0}^{M}p_{ij}^{(2)}u_j^{k+1}+bd\sum_{j=0}^{M}p_{ij}^{(1)}u_j^{k+1}+cdu_i^{k+1}+df_i^{k+1}. \tag{2.5}$$

Let

$$\mathbf{X} = \begin{bmatrix} p_{00}^{(1)} & p_{01}^{(1)} & \cdots & p_{0M}^{(1)} \\ p_{10}^{(1)} & p_{11}^{(1)} & \cdots & p_{1M}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ p_{M0}^{(1)} & p_{M1}^{(1)} & \cdots & p_{MM}^{(1)} \end{bmatrix}, \mathbf{Y} = \begin{bmatrix} p_{00}^{(2)} & p_{01}^{(2)} & \cdots & p_{0M}^{(2)} \\ p_{10}^{(2)} & p_{11}^{(2)} & \cdots & p_{1M}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ p_{M0}^{(2)} & p_{M1}^{(2)} & \cdots & p_{MM}^{(2)} \end{bmatrix}$$

be the weights corresponding to the first and second order derivatives, respectively. We can represent the discretization scheme (2.5) in matrix form as follows:

$$\mathbf{L}\mathbf{U}^1 = \mathbf{U}^0 + d\mathbf{F}^1, \tag{2.6}$$

for $k \geq 1$,

$$\mathbf{L}\mathbf{U}^{k+1} = \sum_{p=0}^{k-1}(\delta_p - \delta_{p+1})\mathbf{U}^{k-p} + \mathbf{U}^0 + d\mathbf{F}^{k+1}, \tag{2.7}$$

where, $\mathbf{I}$ be the identity matrix $(M+1)\times(M+1)$ and

$$\begin{aligned} \mathbf{L} &= (1+cd)\mathbf{I} - \frac{1}{h^2}ad\mathbf{Y} - \frac{1}{h}bd\mathbf{X}, \\ \mathbf{U}^{k+1} &= [u_0, \quad u_1, \quad \ldots, \quad u_M]^T, \quad u_i = u(s_i, \eta_{k+1}), \\ \mathbf{F}^{k+1} &= [f_0, \quad f_1, \quad \ldots, \quad f_M]^T, \quad f_i = f(s_i, \eta_{k+1}). \end{aligned}$$

2.2. **Modified Splines and Calculation of Weights.** This subsection deals with the cubic B-splines and their modifications to find the differential quadrature weights, which is discussed in [8]. For each $j \in \mathbb{Z}$ the standard cubic B-spline $C_j(s)$ which is symmetric about the node point s_j is defined by

$$C_j(s) := \frac{1}{h^3}\begin{cases} (s-s_{j-2})^3, & s \in [s_{j-2}, s_{j-1}), \\ (s-s_{j-2})^3 - 4(s-s_j)^3, & s \in [s_{j-1}, s_j), \\ (s_{j+2}-s)^3 - 4(s_{j+1}-s)^3, & s \in [s_j, s_{j+1}), \\ (s_{j+2}-s)^3, & s \in [s_{j+1}, s_{j+2}), \\ 0, & \text{otherwise.} \end{cases} \tag{2.8}$$

Note that in the interval $[s_{j-2}, s_{j+2}]$, C_j is a twice continuously differentiable function. Table 1 explicitly gives the function values of C_j's and its derivatives at each node point.

TABLE 1. Derivatives of cubic splines at each node.

s	s_{j-2}	s_{j-1}	s_j	s_{j+1}	s_{j+2}
$C_j(x)$	0	1	4	1	0
$C_j'(x)$	0	$\frac{3}{h}$	0	$-\frac{3}{h}$	0
$C_j''(x)$	0	$\frac{6}{h^2}$	$-\frac{12}{h^2}$	$\frac{6}{h^2}$	0

Observe that the cubic splines $C_{-1}, C_0, C_1, C_{M-1}, C_M, C_{M+1}$ are not fully supported in the spatial domain $[a,b]$. Therefore, such standard cubic splines near the boundary must be modified. An optimally accurate modification has

been proposed in [8] and is as follows:

$$\begin{aligned}
\tilde{C}_0 &:= C_0 + 4C_{-1},\\
\tilde{C}_1 &:= C_1 - \tfrac{7}{2}C_{-1} + \tfrac{5}{8}C_0,\\
\tilde{C}_2 &:= C_2 + \tfrac{88}{37}C_{-1} - \tfrac{21}{37}C_0 - \tfrac{4}{37}C_1,\\
\tilde{C}_3 &:= C_3 - C_{-1} + \tfrac{1}{4}C_0 - \tfrac{1}{4}C_2,\\
\tilde{C}_j &:= C_j, \qquad \text{for } j = 4, 5, \ldots, M-4,\\
\tilde{C}_{M-3} &:= C_{M-3} - C_{M+1} + \tfrac{1}{4}C_M - \tfrac{1}{4}C_{M-2},\\
\tilde{C}_{M-2} &:= C_{M-2} + \tfrac{88}{37}C_{M+1} - \tfrac{21}{37}C_M - \tfrac{4}{37}C_{M-1},\\
\tilde{C}_{M-1} &:= C_{M-1} - \tfrac{7}{2}C_{M+1} + \tfrac{5}{8}C_M, \text{ and}\\
\tilde{C}_M &:= C_M + 4C_{M+1}.
\end{aligned}$$

Using these modified splines, we get the matrix equation: $\mathbf{A}\mathbf{X}^T = \mathbf{B}$, where the matrices $\mathbf{A}$ and $\mathbf{B}$ are $(M+1)$ order matrices as follows below: The coefficient matrix $\mathbf{A}$ is invertible; therefore, the weights associated with (2.2) can be calculated from the matrix equation: $\mathbf{X}^T = \mathbf{A}^{-1}\mathbf{B}$. The weights for the second-order derivative (2.3) are given by the matrix $\mathbf{Y}$, obtained by squaring the matrix $\mathbf{X}$, which contains the weights for the first-order derivative [52]. The comprehensive computational procedure for solving the TFBSE (1.5) using the modified cubic B-spline based DQM combined with the $L1$ time discretization is summarized below as Algorithm 1.

$$\mathbf{A} = \begin{bmatrix}
8 & 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0\\
0 & \frac{37}{8} & 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0\\
0 & 0 & \frac{144}{37} & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0\\
0 & 0 & 0 & \frac{15}{4} & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 1 & 4 & 1 & 0 & \cdots & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 1 & 4 & 1 & \cdots & 0 & 0 & 0 & 0\\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots\\
0 & 0 & 0 & 0 & 0 & \cdots & 1 & 4 & 1 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & \frac{15}{4} & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & \frac{144}{37} & 0 & 0\\
0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 & \frac{37}{8} & 0\\
0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 1 & 8
\end{bmatrix},$$

$$\mathbf{B} = \frac{1}{h}\begin{bmatrix}
-12 & -3 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0\\
\frac{27}{2} & -\frac{15}{8} & -3 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0\\
-\frac{276}{37} & \frac{174}{37} & \frac{12}{37} & -3 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0\\
3 & -\frac{3}{2} & 3 & \frac{3}{4} & -3 & 0 & 0 & \cdots & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 3 & 0 & -3 & 0 & \cdots & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 3 & 0 & -3 & \cdots & 0 & 0 & 0 & 0\\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots\\
0 & 0 & 0 & 0 & 0 & \cdots & 3 & 0 & -3 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & \cdots & 0 & 3 & -\frac{3}{4} & -3 & \frac{3}{2} & -3\\
0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 3 & -\frac{12}{37} & -\frac{174}{37} & \frac{276}{37}\\
0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 3 & \frac{15}{8} & -\frac{27}{2}\\
0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 3 & 12
\end{bmatrix}.$$

3. Theoretical Stability

Consider the discretization operator $\mathcal{I}_h$, where h represents the mesh parameter involved. We examine the stability of the numerical scheme associated with $\mathcal{I}_h$ in the sense that it is stable whenever $\mathcal{I}_h^{-1}$ is uniformly bounded [53]. Namely, there exists a constant $\mathcal{C}$ independent of the representative mesh parameter h, with

$$\left\|\mathcal{I}_h^{-1}\right\|_\infty \leq \mathcal{C}.$$

Algorithm 1 MCB-DQM–L1 scheme for the time-fractional Black–Scholes equation.

Initialize the financial parameters r, σ, α, T
Choose the truncated spatial interval $[B_x, B_y]$
Choose number of spatial points M and time steps N
Form the spatial mesh:

$$s_i = B_x + ih, \quad h = \frac{B_y - B_x}{M}, \quad i = 0, 1, \ldots, M.$$

Form the temporal mesh:

$$\eta_k = k\Delta\eta, \quad \Delta\eta = \frac{T}{N}, \quad k = 0, 1, \ldots, N.$$

Construct standard cubic B-spline basis functions
Update boundary splines according to Section 2.2
Generate the matrices A and B
Compute first-order DQM weights:

$$X^T = A^{-1}B$$

Compute second-order weights:

$$Y = X^2$$

Compute fractional coefficients:

$$\delta_p = (p+1)^{1-\alpha} - p^{1-\alpha}$$

Compute $d = \Delta\eta^{\alpha}\Gamma(2-\alpha)$
Assemble system matrix:

$$L = (1+cd)I - \frac{ad}{h^2}Y - \frac{bd}{h}X$$

Set initial condition:

$$U^0 = [u(s_0, 0), \ldots, u(s_M, 0)]^T$$

for $k = 0$ to $N-1$ **do**
    if $k = 0$ **then**
        $RHS = U^0 + dF^1$
    else
        $RHS = \sum_{p=0}^{k-1}(\delta_p - \delta_{p+1})U^{k-p} + U^0 + dF^{k+1}$
    end if
    Impose boundary conditions
    Solve linear system:

$$LU^{k+1} = RHS$$

end for
Return U^k for $k = 0, 1, \ldots, N$

Theorem 3.4 establishing the maximum norm stability of the scheme (2.7).

Lemma 3.1. *[56] Assume A is diagonally dominant by rows and set*

$$\beta = min_k(|a_{kk}| - \sum_{i \neq k} |a_{kj}|).$$

Then

$$\|A^{-1}\|_{\infty} < \frac{1}{\beta}.$$

Lemma 3.2. *Let a, b, c and d be the parameters defined in (1.5,2.5), and let*

$$\mathbf{P} = \frac{a}{h^2}\mathbf{Y} + \frac{b}{h}\mathbf{X},$$

where h, is the spatial mesh parameter. Then,

$$\|\mathbf{P}\|_\infty \leq \frac{|a|}{h^2} \cdot R_Y + \frac{|b|}{h} \cdot R_X,$$

for some positive constants R_X and R_Y.

Proof. Under the assumptions on $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{M+1\times M+1}$, suggested by Lemma 3.1 and the submultiplicative property of matrix norms, we can combine the bounds as follows:

$$R_X = \frac{1}{\beta} \cdot \|\mathbf{B^T}\|_\infty \geq \left\|(\mathbf{A^T})^{-1}\right\|_\infty \|\mathbf{B^T}\|_\infty \geq \left\|(\mathbf{A}^{-1}\mathbf{B})^T\right\|_\infty,$$

where,

$$\beta = \min_k \left(|a_{kk}| - \sum_{l\neq k} |a_{lk}| \right),$$

is the minimum dominance associated with the matrix $\mathbf{A^T}$, and $\|\mathbf{B^T}\|_\infty$ can be estimated numerically by finding the 1-norm of $\mathbf{B} = (b_{ij}) \in \mathbb{R}^{M+1\times M+1}$, for any size $M \geq 8$. Then, by submultiplicativity:

$$R_Y = R_X^2 \geq \|\mathbf{X}\|_\infty^2 \geq \left\|\mathbf{X}^2\right\|_\infty,$$

and, consequently,

$$\|\mathbf{P}\|_\infty = \left\|\frac{a}{h^2}\mathbf{Y} + \frac{b}{h}\mathbf{X}\right\|_\infty \leq \left\|\frac{a}{h^2}\mathbf{Y}\right\|_\infty + \left\|\frac{b}{h}\mathbf{X}\right\|_\infty = \frac{|a|}{h^2} \cdot R_Y + \frac{|b|}{h} \cdot R_X \tag{3.1}$$

□

Lemma 3.3. *(Neumann Series Theorem [49, 51]). Let $\mathscr{X}$ be a Banach space, and $A \in CL(\mathscr{X})$, the space of all continuous linear operators on $\mathscr{X}$, be such that $\|A\|_\infty < 1$. Then*

(1) $I - A$ is invertible in $CL(\mathscr{X})$,

(2) $(I - A)^{-1} = I + A + \cdots + A^n + \cdots = \sum_{n=0}^{\infty} A^n$,

(3) $\left\|(I - A)^{-1}\right\|_\infty \leqslant \dfrac{1}{1 - \|A\|_\infty}$.

In particular, $I - A : \mathscr{X} \to \mathscr{X}$ is bijective: for each $y \in \mathscr{X}$, there exists a unique solution $x \in \mathscr{X}$ of the equation $x - Ax = y$, and, moreover,

$$\|x\|_\infty \leq \left(\frac{1}{1 - \|A\|_\infty}\right) \|y\|_\infty,$$

so that x continuously depends on y.

Proof. Since $\|A^n\|_\infty \leq \|A\|_\infty^n$ and $\|A\|_\infty < 1$, the sequence of partial sums

$$S_n = I + A + A^2 + \cdots + A^n \tag{3.2}$$

forms a Cauchy sequence in $CL(\mathscr{X})$. As $CL(\mathscr{X})$ is also complete, there exists $S \in \mathscr{X}$ such that $S_n \to S$ as $n \to \infty$. To show that S is the inverse of $I - A$, observe that

$$\begin{aligned} S_n(I - A) &= (I + A + A^2 + \cdots + A^n)(I - A) \\ &= I - A^{n+1} \\ &= (I - A)S_n. \end{aligned}$$

Since $A^n \to 0$ as $n \to \infty$, passing to the limit yields

$$S(I-A) = (I-A)S = I,$$

hence $S = (I-A)^{-1}$. Furthermore, using (3.2), we obtain

$$\begin{aligned}\|S - I - A\|_\infty &= \|A^2 + A^3 + \cdots\|_\infty \\ &\leq \sum_{n=2}^{\infty} \|A^n\|_\infty \\ &= \frac{\|A\|_\infty^2}{1 - \|A\|_\infty}.\end{aligned}$$

This completes the proof. □

Theorem 3.4. *Let* $\mathbf{L}$ *be the discrete operator defined by equation (2.7) as*

$$\mathbf{L}(i)\,\boldsymbol{U}^{k+1} = \sum_{p=0}^{k-1} (\delta_p - \delta_{p+1})\,\boldsymbol{U}^{k-p}(i) + \boldsymbol{U}^0(i) + d\boldsymbol{F}^{k+1}(i), \tag{3.2}$$

where, $\mathbf{L}(i)$ *and* $\mathbf{F}(i)^{k+1}$*, denote the* i *th row of* $\mathbf{L}$ *and* $\mathbf{F}^{k+1}$ *respectively, where* $\boldsymbol{U}^{k-p}(i)$ *denotes the* i *th coordinate of* $\boldsymbol{U}^{k-p}$*. Then the discrete operator* $\mathbf{L}$ *satisfies the maximum norm estimate*

$$\|\mathbf{L}^{-1}\|_\infty \leq \mathcal{C},$$

for some constant $\mathcal{C} > 0$*, provides an upper bound independent of the mesh parameter* h*.*

Proof. Let $\mathbf{P} = \frac{a}{h^2}\mathbf{Y} + \frac{b}{h}\mathbf{X}$, then $\mathbf{L}$ can be expressed as $\mathbf{L} = (1+cd)\,\mathbf{I} - d\,\mathbf{P}$. Now, as $c > 0$, we factor the scalar $1+cd$ to obtain

$$\mathbf{L} = (1+cd)\left[\mathbf{I} - \frac{d}{1+cd}\mathbf{P}\right].$$

Let $\mathbf{Q} = \dfrac{d}{1+cd}\mathbf{P}$, so that $\mathbf{L} = (1+cd)(\mathbf{I}-\mathbf{Q})$. Subject to the sufficient conditions on the mesh parameters h and $\Delta\eta$ described by $d\|\mathbf{P}\|_\infty < 1+cd$, we ensure the matrix norm $\|\mathbf{Q}\|_\infty < 1$. Then $\mathbf{I}-\mathbf{Q}$ is invertible by Lemma 3.3, and its inverse is given by the Neumann series:

$$(\mathbf{I}-\mathbf{Q})^{-1} = \sum_{k=0}^{\infty} \mathbf{Q}^k.$$

Moreover, we have the bound (by the Neumann series convergence estimate):

$$\left\|(\mathbf{I}-\mathbf{Q})^{-1}\right\|_\infty \leq \frac{1}{1 - \|\mathbf{Q}\|_\infty}.$$

It follows that $\mathbf{L}$ is invertible and its inverse is given by $\mathbf{L}^{-1} = \frac{1}{1+cd}(\mathbf{I}-\mathbf{Q})^{-1}$. Taking norms, we obtain the bound

$$\left\|\mathbf{L}^{-1}\right\|_\infty \leq \frac{1}{|1+cd|}\left\|(\mathbf{I}-\mathbf{Q})^{-1}\right\|_\infty \leq \frac{1}{|1+cd| - |d|\cdot\|\mathbf{P}\|_\infty} \leq \mathcal{C}.$$

□

Clearly, Theorem (3.4) holds for sufficiently smaller mesh parameters h. Therefore, the uniform boundedness of the inverse operator defined by (3.2) indicates the maximum norm stability of the numerical scheme (2.7) under reasonable conditions on mesh parameters while keeping arbitrary volatility and the rate of returns.

TABLE 2. DQM Error values and OC_s for different α values of Example 4.1.

M	$\alpha = 0.3$				$\alpha = 0.5$			
	L_2 Error	OC_s	L_∞ Error	OC_s	L_2 Error	OC_s	L_∞ Error	OC_s
$2^0 \cdot 10$	$1.330e{-3}$		$7.293e{-4}$		$3.618e{-3}$		$1.779e{-3}$	
$2^1 \cdot 10$	$4.079e{-5}$	5.02	$1.723e{-5}$	5.40	$1.666e{-4}$	4.44	$6.259e{-5}$	4.82
$2^2 \cdot 10$	$1.189e{-6}$	5.10	$3.696e{-7}$	5.54	$7.430e{-6}$	4.49	$2.039e{-6}$	4.94
$2^3 \cdot 10$	$3.421e{-8}$	5.12	$5.191e{-9}$	6.15	$3.307e{-7}$	4.49	$6.566e{-8}$	4.96

M	$\alpha = 0.7$				$\alpha = 0.9$			
	L_2 Error	OC_s	L_∞ Error	OC_s	L_2 Error	OC_s	L_∞ Error	OC_s
$2^0 \cdot 10$	$8.308e{-3}$		$3.514e{-3}$		$1.758e{-2}$		$7.813e{-3}$	
$2^1 \cdot 10$	$5.945e{-4}$	3.80	$1.846e{-4}$	4.25	$1.975e{-3}$	3.15	$6.415e{-4}$	3.61
$2^2 \cdot 10$	$4.175e{-5}$	3.83	$9.357e{-6}$	4.30	$2.202e{-4}$	3.16	$5.115e{-5}$	3.65
$2^3 \cdot 10$	$2.939e{-6}$	3.83	$4.696e{-7}$	4.32	$2.463e{-5}$	3.16	$4.064e{-6}$	3.65

4. Numerical Findings

In this section, two examples with exact solutions are provided to illustrate the precision of the solution and the order of convergence of the suggested numerical approach. Let u_j and $\tilde{u}_j$ be the exact and numerical solution at the node point j respectively. The L_2 norm error is evaluated using

$$L_2 = \sqrt{h \sum_{j=0}^{M} \left| u_j - \tilde{u}_j \right|^2}, \tag{4.1}$$

and L_∞ : the maximum norm error by

$$L_\infty = \max_{0 \le j \le M} \left| u_j - \tilde{u}_j \right|. \tag{4.2}$$

Furthermore, for evaluating the spatial order of convergence OC_s

$$OC_s = \frac{\log(E^{h_1}/E^{h_2})}{\log(h_1/h_2)}, \tag{4.3}$$

where, E^{h_1} and E^{h_2} represent the errors in the size of the space mesh h_1 and h_2, respectively, and for the order of convergence OC_η in time

$$OC_\eta = \frac{\log(E^{k_1}/E^{k_2})}{\log(k_1/k_2)}, \tag{4.4}$$

where, E^{k_1} and E^{k_2} represent errors in time step sizes k_1 and k_2, respectively. All Computations are done on an Apple M2 Max 64GB MacOS 15.6.1 operating system using Python coding.

Example 4.1. Here, we consider the time fractional BS equation, which takes the following form:

$$\begin{cases} {}_0D_\eta^\alpha u(s,\eta) = a\dfrac{\partial^2 u(s,\eta)}{\partial s^2} + b\dfrac{\partial u(s,\eta)}{\partial s} - cu(s,\eta) + f(s,\eta), \\ u(0,\eta) = 0, \quad u(1,\eta) = 0, \\ u(s,0) = s^2(1-s), \end{cases} \tag{4.5}$$

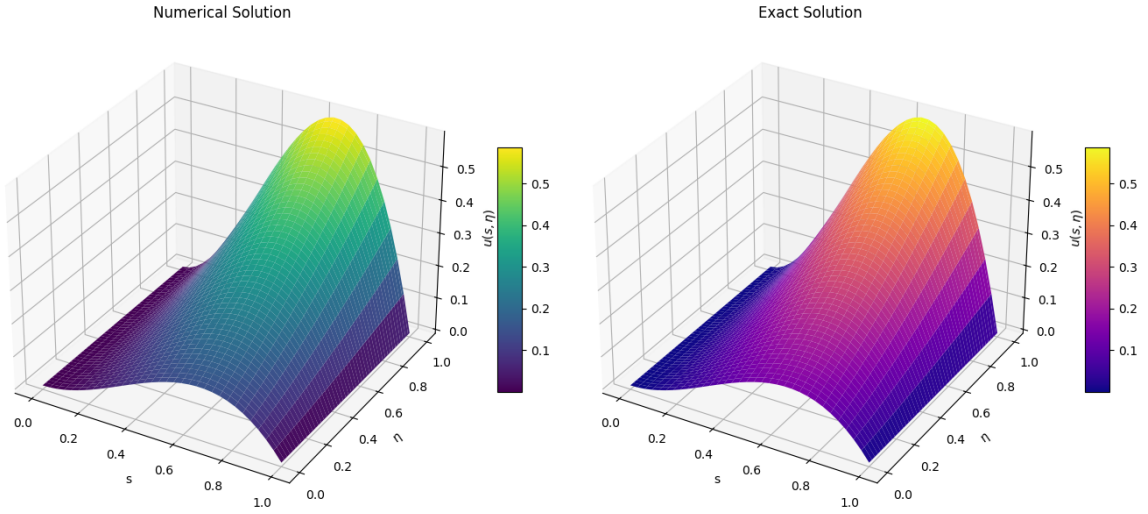


FIGURE 1. Surface plot of numerical and exact solution at $\alpha = 0.7, M = 20$ of Example 4.1.

TABLE 3. Error values and OC_η for different α values of Example 4.1.

M	$\alpha = 0.3$				$\alpha = 0.5$			
	L_2 Error	OC_η	L_∞ Error	OC_η	L_2 Error	OC_η	L_∞ Error	OC_η
$2^0 \cdot 10$	$3.375e{-3}$		$7.293e{-4}$		$9.372e{-3}$		$1.779e{-3}$	
$2^1 \cdot 10$	$1.100e{-3}$	1.62	$2.404e{-4}$	1.60	$3.436e{-3}$	1.44	$6.598e{-4}$	1.43
$2^2 \cdot 10$	$3.540e{-4}$	1.64	$7.789e{-5}$	1.63	$1.244e{-3}$	1.47	$2.408e{-6}$	1.45
$2^3 \cdot 10$	$1.127e{-4}$	1.65	$2.492e{-5}$	1.64	$4.469e{-4}$	1.48	$8.701e{-5}$	1.47

M	$\alpha = 0.7$				$\alpha = 0.9$			
	L_2 Error	OC_η	L_∞ Error	OC_η	L_2 Error	OC_η	L_∞ Error	OC_η
$2^0 \cdot 10$	$6.142e{-1}$		$1.221e{-1}$		$1.38e0$		$2.656e{-1}$	
$2^1 \cdot 10$	$2.567e{-1}$	1.26	$5.099e{-2}$	1.26	$6.601e{-1}$	1.06	$1.261e{-1}$	1.07
$2^2 \cdot 10$	$1.068e{-1}$	1.27	$2.113e{-2}$	1.27	$3.109e{-1}$	1.09	$5.933e{-2}$	1.09
$2^3 \cdot 10$	$4.365e{-2}$	1.29	$8.639e{-3}$	1.29	$1.459e{-1}$	1.09	$2.782e{-2}$	1.09

where the source term

$$f(s,\eta) = \left(\frac{2\eta^{2-\alpha}}{\Gamma(3-\alpha)} + \frac{2\eta^{1-\alpha}}{\Gamma(2-\alpha)}\right) s^2(1-s) - (\eta+1)^2\left[a(2-6s) + b\left(2s-3s^2\right) - cs^2(1-s)\right],$$

is selected so that the exact solution is given by [59, 61],

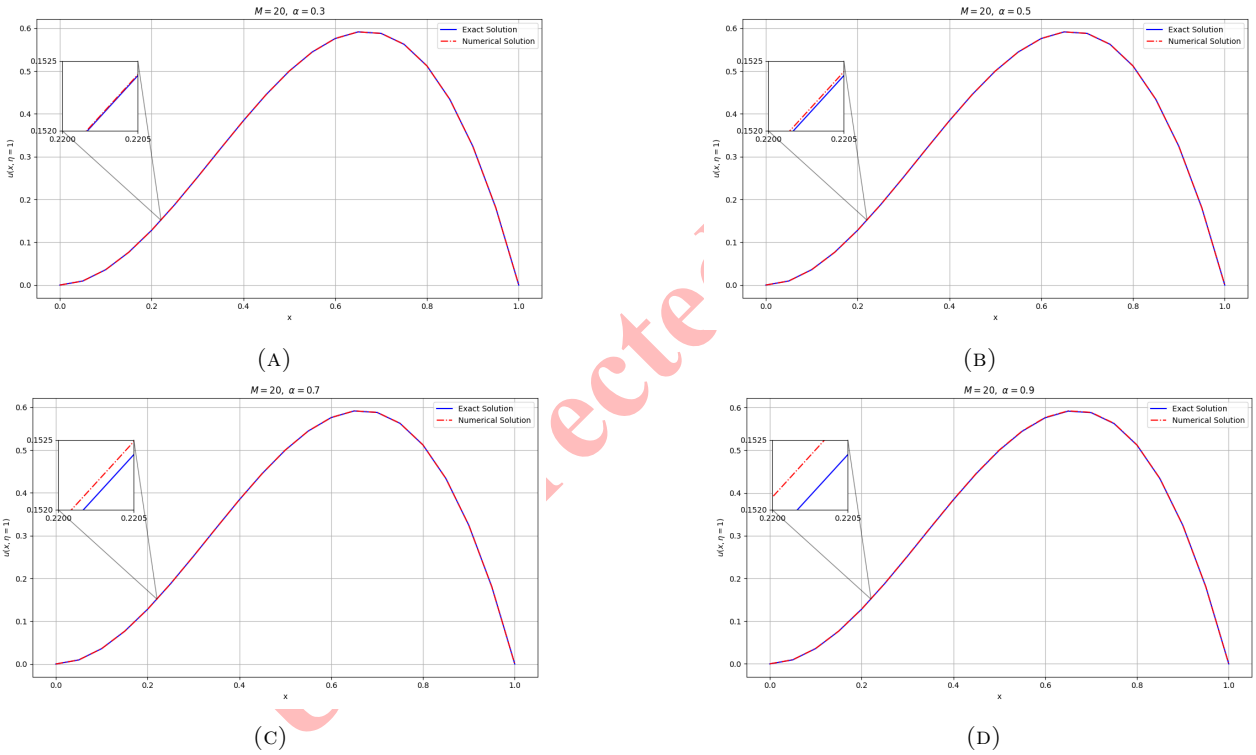
$$u(s,\eta) = (\eta+1)^2 s^2(1-s).$$

The parameter values are considered as

$$r = 0.05,\ \sigma = 0.25,\ a = \frac{1}{2}\sigma^2,\ b = r - a,\ c = r \ \text{ and } \ T = 1.$$

The numerical error values and spatial order of convergence (OC_s) for $M = 2^m \cdot 10,\ N = 10^{m+1},\ m = 0, 1, 2, ...,$ are displayed in Table 2. The tabulated results demonstrate that the scheme achieves high-order accuracy in both L_2 and

L_∞ norms. The high order of convergence here shows the efficiency of the modified spline-based differential quadrature method. Moreover, the results for different values of α indicate the memory effect in financial markets. Whenever α tends to zero the error is decreasing and have high order of convergence. Also if α tends to one, then the model approaches to the classical Black-Scholes behavior and have an order of convergence four. From a realistic perspective, this indicates that the fractional models are crucial in real-world financial markets, which allows the model to capture the memory and realistic market phenomena such as long-term memory dependence and volatility clustering. Table 3 displays the error values and order of convergence in time for $M = 80, \quad N = 2^m \cdot 10, \ m = 0, 1, 2, ...$ The data gives the order of convergence in time is $2 - \alpha$. The surface plot of the solutions when $\alpha = 0.7$, $M = 20$ is given in Figure 1. Figures 2:(A)-(D) show the numerical and exact solutions at the payoff for different α values with $M = 20$. The absolute error values are showed in Figure 3 (A)-(D). Moreover, H. Zhang [61] formed an implicit discrete scheme for the considered model by the combination of finite difference method(FDM) to the space derivatives and classical L1 scheme for the time fractional derivative. The numerical data in [61] shows that the scheme has spatial order of convergence two. For $M = 8, N = 1000$, the results from [61]



(A) (B) (C) (D)

FIGURE 2. (A)-(D) Comparison of Numerical vs Exact solution for different values of α at $M = 20$ of Example 4.1.

show an L_2 error $6.1678e^{-4}$ and L_∞ error $7.6750e^{-4}$.In comparison, the proposed method yields an L_2 error $2.019e^{-5}$ and L_∞ error $9.286e^{-6}$ for for the same parameters. The tabulated CPU time (in seconds) is given in Table 4. The numerical error values for this FDM scheme [61] for different values of α at $M = 2^m \cdot 10, \quad N = 10^{m+1}, \ m = 0, 1, 2, ...,$ are displayed in Table 5. The mixed alternate segment Crank-Nicolson (MASC-N) scheme developed by X. Yang [59] has spatial order of convergence two. On Comparing with these techniques, the results shows the potential of modified B-spline based DQM for solving fractional differential equations with precision and stability.

TABLE 4. CPU time (in seconds) for different values of M and fractional orders α for Example 4.1.

$\alpha = 0.5$		$\alpha = 0.7$	
M	CPU time (in seconds)	M	CPU time (in seconds)
$2^0 \cdot 10$	$6.30e^{-4}$	$2^0 \cdot 10$	$6.30e^{-4}$
$2^1 \cdot 10$	$8.76e^{-3}$	$2^1 \cdot 10$	$8.78e^{-3}$
$2^2 \cdot 10$	$6.945e^{-1}$	$2^2 \cdot 10$	$6.92e^{-1}$

TABLE 5. FDM Error values and OC_s for different α values of Example 4.1.

M	$\alpha = 0.3$				$\alpha = 0.5$			
	L_2 Error	OC_s	L_∞ Error	OC_s	L_2 Error	OC_s	L_∞ Error	OC_s
$2^0 \cdot 10$	$3.523e{-4}$		$2.085e{-4}$		$2.605e{-3}$		$1.284e{-3}$	
$2^1 \cdot 10$	$4.799e{-4}$	-0.44	$1.346e{-4}$	0.631	$3.296e{-4}$	2.98	$9.477e{-5}$	3.76
$2^2 \cdot 10$	$1.823e{-4}$	1.39	$3.635e{-5}$	1.89	$1.650e{-4}$	0.99	$3.256e{-5}$	1.54
$2^3 \cdot 10$	$6.486e{-5}$	1.49	$9.147e{-6}$	1.99	$6.059e{-5}$	1.45	$8.483e{-6}$	1.94

M	$\alpha = 0.7$				$\alpha = 0.9$			
	L_2 Error	OC_s	L_∞ Error	OC_s	L_2 Error	OC_s	L_∞ Error	OC_s
$2^0 \cdot 10$	$3.523e{-4}$		$2.084e{-4}$		$1.812e{-2}$		$8.538e{-3}$	
$2^1 \cdot 10$	$1.979e{-4}$	0.83	$7.503e{-5}$	1.47	$1.611e{-3}$	3.49	$5.509e{-4}$	3.95
$2^2 \cdot 10$	$1.191e{-4}$	0.73	$2.363e{-5}$	1.67	$9.230e{-5}$	4.12	$2.480e{-5}$	4.47
$2^3 \cdot 10$	$5.352e{-5}$	1.15	$7.385e{-6}$	1.67	$2.806e{-5}$	1.71	$4.436e{-6}$	2.48

Example 4.2. In this example we consider the time fractional BS equation with non homogeneous boundary condition having the form:

$$\begin{cases} {}_0D_\eta^\alpha u(s,\eta) = a\dfrac{\partial^2 u(s,\eta)}{\partial s^2} + b\dfrac{\partial u(s,\eta)}{\partial s} - cu(s,\eta) + f(s,\eta), \\ u(0,\eta) = (\eta+1)^2, \quad u(1,\eta) = 3(\eta+1)^2, \\ u(s,0) = s^3 + s^2 + 1, \end{cases} \tag{4.6}$$

where the source term $f(s,\eta)$ is

$$\left(\frac{2\eta^{2-\alpha}}{\Gamma(3-\alpha)} + \frac{2\eta^{1-\alpha}}{\Gamma(2-\alpha)}\right)(s^3+s^2+1) - (\eta+1)^2\left[a(6s+2) + b\left(2s+3s^2\right) - c(s^3+s^2+1)\right],$$

is chosen so that the exact solution is given by [59, 61]

$$u(s,\eta) = (\eta+1)^2(s^3+s^2+1).$$

Here we take the parameters values as

$$r = 0.5, \quad a = 1, \quad b = r - a, \quad c = r \text{ and, } T = 1.$$

The evaluated numerical error values and spatial order of convergence (OC_s) for $M = 2^m \cdot 10$, $N = 10^{m+1}, m = 0, 1, 2, \ldots$ are displayed in Table 6. The tabulated results indicate that for all values of α, the errors in both norms decreases as M increases, indicating that the numerical method converges with refinement. We observe that from the Table 6, for α tending to zero, the order of convergence is relatively high, shows the memory effect of the model,

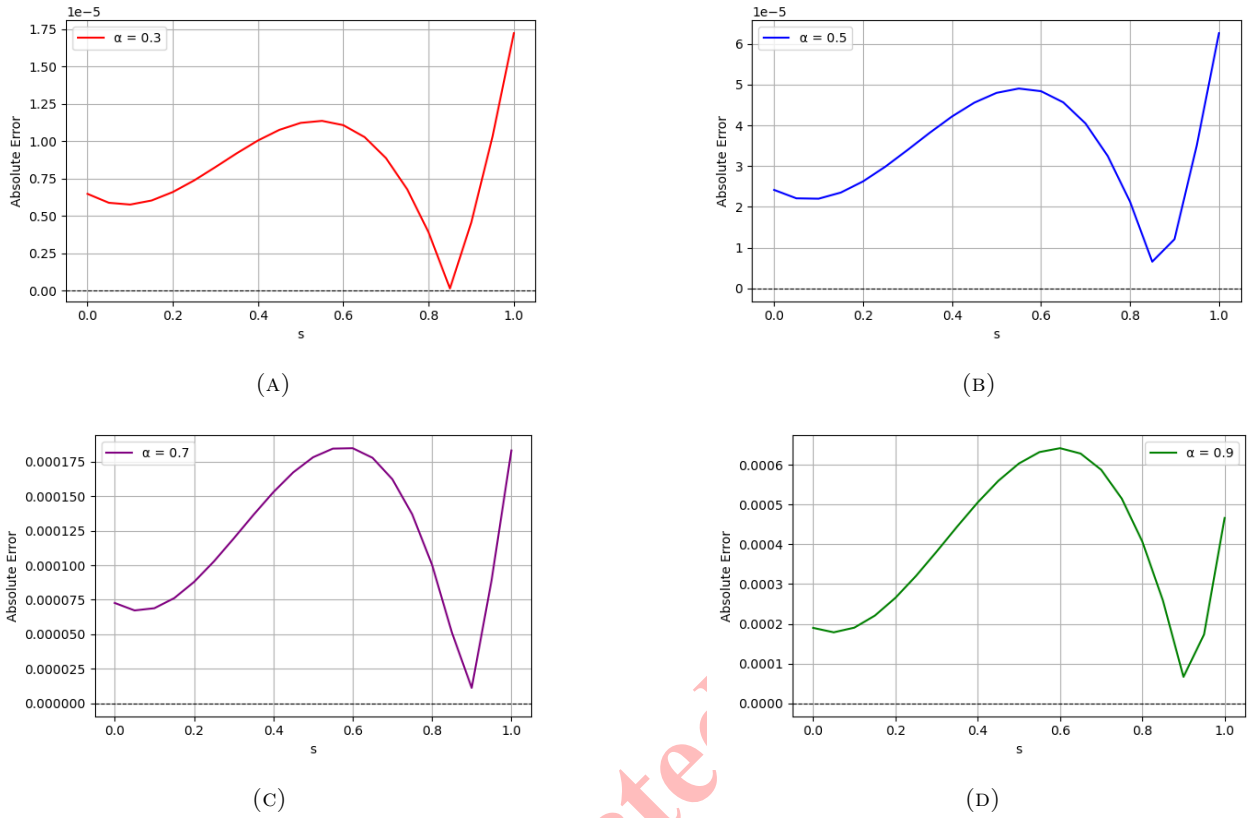


FIGURE 3. (A)-(D) Absolute Error plot for different values of α at $M = 20$ of Example 4.1.

when α goes to one the order of convergence is near to 4, compared with the existing method in [61]. As discussed in the Example 4.1, the numerical values for different α suggest that the fractional model can be effectively applied in practical scenarios such as option pricing with memory, where the classical model fails to provide these predictions. Table 7 displays the error values and the order of convergence in time for $M = 80, \quad N = 2^m \cdot 10, \ m = 0, 1, 2, ...,$ and the data demonstrate that the order of convergence in time is $2 - \alpha$. The CPU time (in seconds) tabulation is given in Table 8. Figure 4 presents the surface plot of the solutions for $\alpha = 0.7$ and $M = 20$. Figure 5: (A)-(D) presents the comparison of numerical and exact solution for various values of α for $M = 20$. The absolute error values are displayed in Figure 6 (A)-(D).

5. Conclusion

In this study, we have developed an implicit robust numerical scheme for solving the TFBSE, which generalizes option pricing with memory and non-local effects. The fractional formulation, governed by a modified Riemann-Liouville derivative of the Caputo type, allows the model to capture long-range dependence and anomalous diffusion phenomena that the classical Black-Scholes equation fails to address. Thus the fractional model considered in the work is a more realistic representation of financial markets by incorporating memory-dependent dynamics. These dynamics are essential for capturing volatility clustering and persistent correlations. The proposed scheme combines L1 finite difference for time discretization and modified cubic B-spline DQM for spatial discretization. The L1 scheme effectively manages the non-local temporal operator while maintaining an accuracy order of $\mathcal{O}(k^{2-\alpha})$, which is ideal for fractional dynamics. On the spatial side, the implementation of modified cubic B-splines, specifically adapted to preserve optimal polynomial reproduction properties throughout the domain, including near boundaries, ensures a

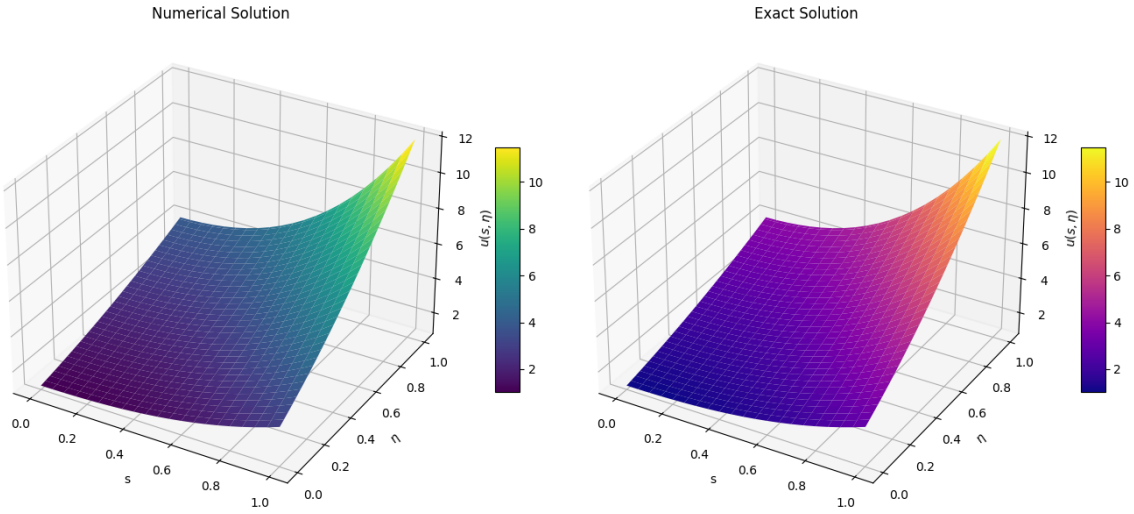


FIGURE 4. Surface plot of numerical and exact solution at $\alpha = 0.7, M = 20$ of Example 4.2.

TABLE 6. Error and OC_s for different α values of Example 4.2.

M	$\alpha = 0.3$				$\alpha = 0.5$			
	L_2 Error	OC_s	L_∞ Error	OC_s	L_2 Error	OC_s	L_∞ Error	OC_s
$2^0 \cdot 10$	$3.104e{-2}$		$1.737e{-2}$		$9.252e{-2}$		$5.034e{-2}$	
$2^1 \cdot 10$	$9.560e{-4}$	5.02	$3.975e{-4}$	5.45	$4.268e{-3}$	4.44	$1.719e{-3}$	4.87
$2^2 \cdot 10$	$2.116e{-5}$	5.46	$6.643e{-6}$	5.90	$1.728e{-4}$	4.63	$5.242e{-5}$	5.03

M	$\alpha = 0.7$				$\alpha = 0.9$			
	L_2 Error	OC_s	L_∞ Error	OC_s	L_2 Error	OC_s	L_∞ Error	OC_s
$2^0 \cdot 10$	$2.328e{-1}$		$1.222e{-1}$		$7.109e{-1}$		$2.656e{-1}$	
$2^1 \cdot 10$	$1.678e{-2}$	3.79	$6.466e{-3}$	4.24	$5.897e{-2}$	3.59	$2.182e{-2}$	3.60
$2^2 \cdot 10$	$1.188e{-3}$	3.82	$3.287e{-4}$	4.07	$6.542e{-3}$	3.17	$1.742e{-3}$	3.64

highly accurate approximation of both first- and second-order derivatives. Numerical experiments validate its superior performance in pricing European options across different fractional order values α. Furthermore, the stability of the method is rigorously analyzed by deriving an upper bound for the maximum norm of the inverse operator. A key analytical contribution of this work lies in the rigorous stability analysis. Thanks to the Neumann series theorem, it provides a uniform bound for the inverse operator under reasonable conditions on the mesh parameters. The performance of the proposed method is evaluated using two test cases with exact solutions. The numerical results demonstrate that the proposed method attains fourth-order convergence in space and convergence order of $2 - \alpha$ in time. Moreover, we observe an enhancement in order of spatial convergence whenever α tends to zero. This indicates the ability of the model to handle the memory effects inherent in fractional derivatives. The numerical results of the proposed scheme demonstrate its efficiency in comparison to existing finite difference-based approaches. The studies in the literature [59, 61] reported second-order convergence in space. On comparing with these methods, the present modified B-spline-based differential quadrature method achieves fourth-order spatial convergence. Furthermore, the CPU time results show that the proposed method provides accurate numerical solutions with reasonable computational

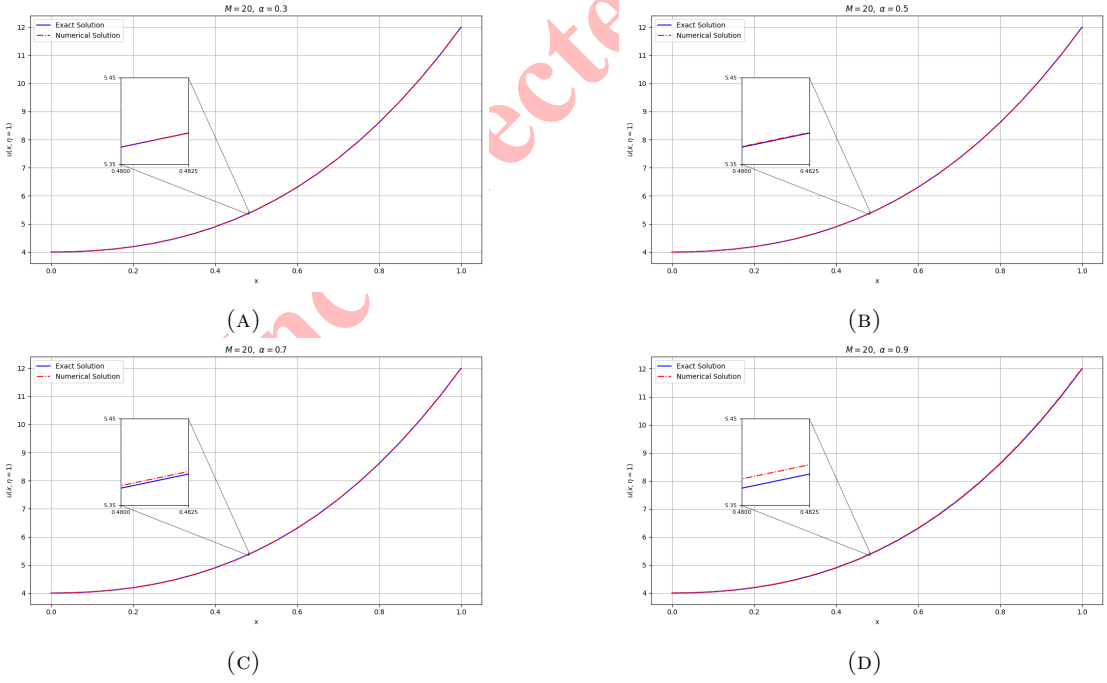
TABLE 7. Error and OC_η for different α values of Example 4.2.

M	$\alpha = 0.3$				$\alpha = 0.5$			
	L_2 Error	OC_η	L_∞ Error	OC_η	L_2 Error	OC_η	L_∞ Error	OC_η
$2^0 \cdot 10$	$7.894e{-2}$		$1.709e{-2}$		$2.415e{-1}$		$5.015e{-2}$	
$2^1 \cdot 10$	$2.470e{-2}$	1.68	$5.442e{-3}$	1.65	$9.045e{-2}$	1.42	$1.862e{-2}$	1.43
$2^2 \cdot 10$	$6.984e{-3}$	1.82	$1.607e{-3}$	1.75	$3.120e{-2}$	1.54	$6.573e{-3}$	1.50

M	$\alpha = 0.7$				$\alpha = 0.9$			
	L_2 Error	OC_η	L_∞ Error	OC_η	L_2 Error	OC_η	L_∞ Error	OC_η
$2^0 \cdot 10$	$6.142e{-1}$		$1.221e{-1}$		$1.38e0$		$2.656e{-1}$	
$2^1 \cdot 10$	$2.567e{-1}$	1.26	$5.099e{-2}$	1.27	$6.601e{-1}$	1.06	$1.261e{-2}$	1.07
$2^2 \cdot 10$	$1.068e{-1}$	1.27	$2.113e{-2}$	1.27	$3.109e{-1}$	1.09	$5.933e{-2}$	1.09

TABLE 8. CPU time (in seconds) for different values of M and fractional orders α of Example 4.2.

$\alpha = 0.5$		$\alpha = 0.7$	
M	CPU time (in seconds)	M	CPU time (in seconds)
$2^0 \cdot 10$	$6.30e^{-4}$	$2^0 \cdot 10$	$6.30e^{-4}$
$2^1 \cdot 10$	$8.70e^{-3}$	$2^1 \cdot 10$	$8.85e^{-3}$
$2^2 \cdot 10$	$6.92e^{-1}$	$2^2 \cdot 10$	$6.95e^{-1}$



(A) (B)

(C) (D)

FIGURE 5. (A)-(D) Comparison of Numerical vs Exact solution for different values of α at $M = 20$ of Example 4.2.

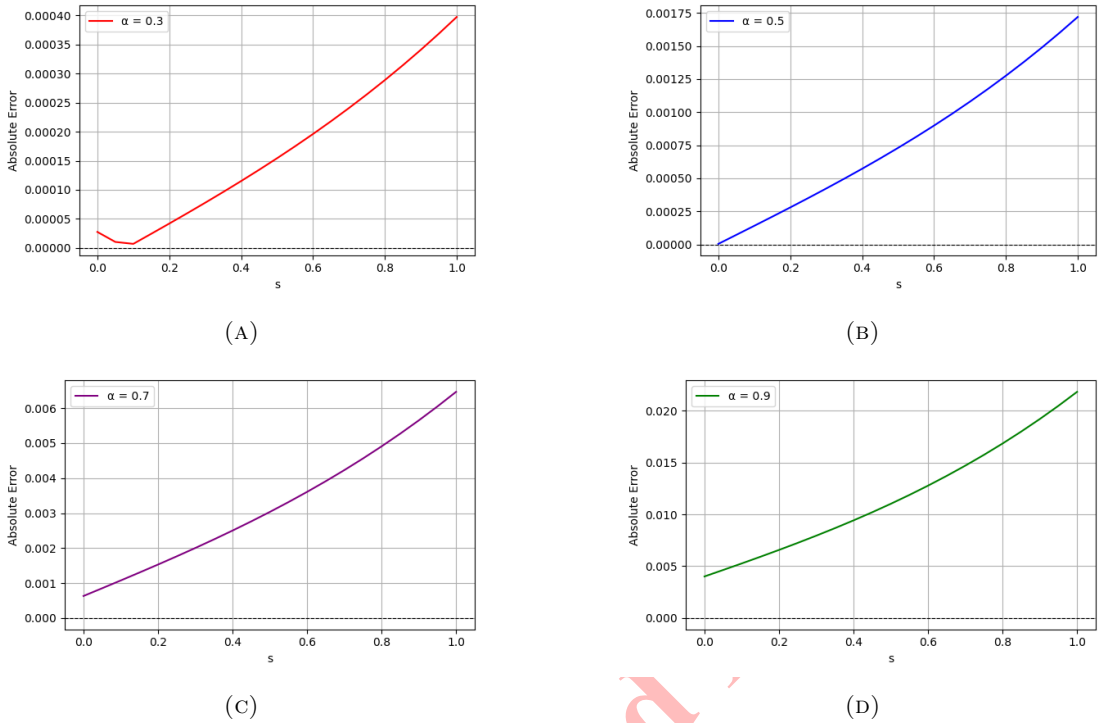


FIGURE 6. (A)-(D) Absolute Error plot for different values of α at $M = 20$ of Example 4.2.

cost. Future work could extend this approach to jump-diffusion models or regime-switching fractional models. Such extensions would enhance the applicability of the proposed approach to more realistic financial markets that exhibit sudden price jumps, memory effects, and regime-dependent volatility. The high accuracy and stability of the method make it a promising tool for fractional financial models, with potential applications in exotic option pricing and risk management.

ACKNOWLEDGMENT

Funding: The authors thank the University Grants Commission (UGC), India, for the financial assistance provided for this research. The Author 3 gratefully acknowledges the financial support provided by the RUSA 2.0/T3A project, Cochin University of Science and Technology.

Disclosure of potential conflicts of interest: The authors declare that they have no conflict of interest.

Authors Contributions: All authors contributed to the conceptualization of the study, the development of the model, the convergence analysis, and the preparation of the manuscript. Each author has reviewed and approved the final version of the manuscript.

Data Availability: Not applicable.

References

[1] S. Abdi-Mazraeh and A. Khani, *An efficient computational algorithm for pricing European, barrier and American options*, Comput. Appl. Math., *37*(4) (2018), 4856-4876.

[2] S. Abdi-Mazraeh, A. Khani, and S. Irandoust-Pakchin, *Multiple Shooting Method for Solving Black–Scholes Equation*, Comput. Econ., *56*(4) (2020), 723–746.

[3] S. Abdi-Mazraeh, H. Kheiri, and S. Irandoust-Pakchin, *Construction of operational matrices based on linear cardinal B-spline functions for solving fractional stochastic integro-differential equation*, J. Appl. Math. Comput., *68*(1) (2022), 151-175.

[4] D. Ahmadian, L. V. Ballestra, and N. Karimi, *An extremely efficient numerical method for pricing options in the Black–Scholes model with jumps*, Math. Methods Appl. Sci., *44*(2) (2021), 1843-1862.

[5] T. Akram, M. Abbas, K.M. Abualnaja, A. Iqbal, and A. Majeed, *An efficient numerical technique based on the extended cubic B-spline functions for solving time fractional Black–Scholes model*, Eng. Comput., *38*(2) (2022), 1705-1716.

[6] V. S. Aswin, T. K. Riyasudheen, and A. Awasthi, *Differential quadrature parallel algorithms for solving systems of convection-diffusion and reaction models*, Numer. Algorithms, *93*(1) (2023), 321-346.

[7] A. Awasthi and T. K. Riyasudheen, *An accurate solution for the generalized Black-Scholes equations governing option pricing*, AIMS Math., *5*(3) (2020), 2226-2243.

[8] A. Babu, B. Han, and N. Asharaf, *Numerical solution of the hyperbolic telegraph equation using cubic B-spline based differential quadrature of high accuracy*, Comput. Methods Differ. Equ., *10*(4) (2022), 837-859.

[9] A. Bashan, N. M. Yagmurlu, Y. Ucar, and A. Esen, *An effective approach to numerical soliton solutions for the Schrödinger equation via modified cubic B-spline differential quadrature method*, Chaos Solitons Fractals, *100* (2017), 45–56.

[10] A. Bashan, *An efficient approximation to numerical solutions for the kawahara equation via modified cubic B-spline differential quadrature method*, Mediterr. J. Math., *16*(1) (2019), 1-19.

[11] A. Bashan, *An effective approximation to the dispersive soliton solutions of the coupled KdV equation via combination of two efficient methods*, Comput. Appl. Math., *39*(2) (2020).

[12] R. Bellman, B. G. Kashef, and J. Casti, *Differential quadrature: A technique for the rapid solution of nonlinear partial differential equations*, J. Comput. Phy., *10*(1) (1972), 40-52.

[13] F. Black and M. Scholes, *The pricing of options and corporate liabilities*, J. Polit. Econ., *81*(3) (1973), 637–654.

[14] P. P. Boyle, *Options: A monte carlo approach*, J. Financ. Econ., *4*(3) (1977), 323–338.

[15] P. P. Boyle, *Approximating American options with the binomial model*, J. Financ. Econ., *4*(4) (1986), 375–387.

[16] A. Cartea and D. Del-Castillo-Negrete, *Fractional diffusion models of option prices in markets with jumps*, Phys. A: Stat. Mech. Appl., *374*(2) (2007), 749–763.

[17] W. Chen, X. Xu, and S. Zhu, *Analytically pricing double barrier options based on a time-fractional Black–Scholes equation*, Comput. Math. Appl., *69*(12) (2015), 1407-1419.

[18] W. Chen and S. Lin, *Option Pricing under the KOBOL Model*, ANZIAM J., *60*(2) (2018), 175-190.

[19] C. W. Cheong, *Self-similarity in financial markets: A fractionally integrated approach*, Math. Comput. Model., *52*(3-4) (2010), 459-471.

[20] P. Christoffersen, S. Heston, and K. Jacobs, *The shape and term structure of the index option smirk: Why multifactor stochastic volatility models work so well*, Manag. Sci., *55*(12) (2009), 1914-1932.

[21] A. A. Elbeleze, A. Kılıçman, and B. M. Taib, *Homotopy Perturbation Method for Fractional Black-Scholes European Option Pricing Equations Using Sumudu Transform*, Math. Prob. Eng., *2013*(2) (2013), 1-7.

[22] S. E. Fadugba, *Homotopy analysis method and its applications in the valuation of European call options with time-fractional Black-Scholes equation*, Chaos Solitons Fractals, *141* (2020), 110351.

[23] H. Ghafouri, M. Ranjbar, and A. Khani, *Application of cubic B-spline quasi-interpolation for solving time fractional partial differential equation*, Comput. Methods Differ. Equ., *8*(4) (2020), 781–793.

[24] J. He and A. Zhang, *Finite difference/fourier spectral for a time fractional Black-Scholes model with option pricing*, Math. Probl. Eng., *2020*(1) (2020), 1-9.

[25] S. L. Heston, *A closed-form solution for options with stochastic volatility with applications to bond and currency options*, Rev. Financ. Stud., *6*(2) (1993), 327–343.
[26] S. Irandoust-Pakchin, *Exact solutions for some of the fractional differential equations by using modification of He's variational iteration method*, Math. Sci., *5*(1) (2011), 51-60.
[27] S. Irandoust-Pakchin, S. Abdi-mazraeh, and H. Kheiri, *Construction of new generating function based on linear barycentric rational interpolation for numerical solution of fractional differential equations*, J. Comput. Appl. Math., *375* (2020), 112799.
[28] S. Irandoust-Pakchin and S. Abdi-Mazraeh, *Fractional second linear multistep methods: the explicit forms for solving fractional differential equations and stability analysis*, Int. J. Comput. Math., *100*(1) (2023), 20-46.
[29] R. Jiwari, *Lagrange interpolation and modified cubic B-spline differential quadrature methods for solving hyperbolic partial differential equations with Dirichlet and Neumann boundary conditions*, Comput. Phys. Commun., *193*(3) (2015), 55-65.
[30] M. K. Kadalbajoo, L. P. Tripathi, and A. Kumar, *A cubic B-spline collocation method for a numerical solution of the generalized Black–Scholes equation*, Math. Comput. Modell., *55*(3-4) (2012), 1483-1505.
[31] M. K. Kadalbajoo, L. P. Tripathi, and P. Arora, *A robust nonuniform B-spline collocation method for solving the generalized Black–Scholes equation*, IMA J. Numer. Anal., *34*(1) (2014), 252–278.
[32] S. B. G. Karakoç, A. Başhan, and T. Geyikli, *Two Different Methods for Numerical Solution of the Modified Burgers' Equation*, Sci World Journal, *2014* (2014), 1-13.
[33] V. S. Kiryakova, *Generalized fractional calculus and applications*, CRC press, Boca Raton, USA, 1993.
[34] A. Korkmaz and I. Dag, *Polynomial based differential quadrature method for numerical solution of nonlinear Burgers' equation*, J. Frankl. Inst., *348*(10) (2011), 2863–2875.
[35] S. G. Kou, *A jump-diffusion model for option pricing*, Manag. Sci., *48*(8) (2002), 1086–1101.
[36] G. Krzyzanowski, M. Magdziarz, and L. A. Plociniczak, *Weighted finite difference method for subdiffusive Black-Scholes model*, Comput. Math. Appl., *80*(5) (2020), 653–670.
[37] W. Li, *The numerical solution of fractional order equation in financial models and its application*, 2009, Thesis (Ph.D.)-Hangzhou University of Electronic Science and Technology-Hangzhou-Zhejiang-China.
[38] Y. Lin and C. Xu, *Finite difference/spectral approximations for the time-fractional diffusion equation*, J. Comput. Phy., *225*(2) (2007), 1533-1552.
[39] C. Lubich, *Discretized fractional calculus*, SIAM J. Math. Anal., *17*(3) (1986), 704–719.
[40] R. C. Merton, *An intertemporal capital asset pricing model*, Econometrica, *41*(5) (1973), 867–887.
[41] R. C. Merton, *Theory of rational option pricing*, Bell J. Econ. Manage. Sci., *4*(1) (1973), 141–183.
[42] R. C. Mittal and R. K. Jain, *Numerical solutions of nonlinear Burgers' equation with modified cubic B-splines collocation method*, Appl. Math. Comput., *218*(5) (2012), 7839-7855.
[43] S. M. Nuugulu, F. Gideon, and K. C. Patidar, *A robust numerical solution to a time-fractional Black–Scholes equation*, Adv. Differ. Equ., *2021*(1) (2021), 123.
[44] E. Panas, *Long memory and chaotic models of prices on the london metal exchange*, Resour. Policy, *27*(4) (2001), 235–246.
[45] P. Phaochoo, A. Luadsong, and N. Aschariyaphotha, *The meshless local Petrov-Galerkin based on moving kriging interpolation for solving fractional Black-Scholes model*, J. King Saud-Univ. Sci., *28*(1) (2016), 111–117.
[46] A. Rathinasamy, D. Ahmadian, and P. Nair, *Second-order balanced stochastic Runge–Kutta methods with multi-dimensional studies*, J. Comput. Appl. Math., *377* (2020), 112890.
[47] M. Rezaei, A. R. Yazdanian, A. Ashrafi, and S. M. Mahmoudi, *Numerically pricing nonlinear time-fractional Black-Scholes equation with time-dependent parameters under transaction costs*, Comput. Econ., *60*(1) (2022), 243–280.
[48] P. Roul, *A high accuracy numerical method and its convergence for time-fractional Black-Scholes equation governing European options*, Appl. Numer. Math., *151* (2020), 472–493.
[49] O. F. Rouz and D. Ahmadian, *Stability analysis of two classes of improved backward Euler methods for stochastic delay differential equations of neutral type*, Comput. Methods Differ. Equ., *5*(3) (2017), 201-213.

[50] A. Safdari-Vaighani, D. Ahmadian, and R. Javid-Jahromi, *An Approximation Scheme for Option Pricing Under Two-State Continuous CAPM*, Comput. Econ., *57*(4) (2021), 1373-1385.
[51] A. Sasane, *A friendly approach to functional analysis*, World Scientific Publishing Europe Ltd., London, 2017.
[52] C. Shu, *Differential Quadrature and Its Application in Engineering*, Springer Science & Business Media, 2000.
[53] R. Teman, *Numerical analysis*, Springer Science & Business Media, 2012.
[54] Z. Tian, S. Zhai, H. Ji, and Z. Weng, *A compact quadratic spline collocation method for the time-fractional Black–Scholes model*, J. Appl. Math. Comput., *66* (2021), 327-350.
[55] R. Valkov, *Fitted finite volume method for a generalized Black–Scholes equation transformed on finite interval*, Numer. Algorithms, *65* (2014), 195–220.
[56] J. M. Varah, *A lower bound for the smallest singular value of a matrix*, Linear Algebra Appl., *11* (1975), 3-5.
[57] S. Wang, *A novel fitted finite volume method for the Black–Scholes equation governing option pricing*, IMA J. Numer. Anal., *24*(4) (2004), 699–720.
[58] W. Wyss, *The fractional Black–Scholes equation*, Fract. Calc. Appl. Anal. Theory Appl., *3*(1) (2000), 51–61.
[59] X. Yang, L. Wu, and Y. Zhang, *A New Parallel Difference Method for Solving Time Fractional Black-Scholes Model*, J. Math. Fin., *12*(4) (2022), 683-701.
[60] J. Yu, Y. Feng, and X. Wang, *Lie symmetry analysis and exact solutions of time fractional Black–Scholes equation*, Int. J. Financ. Engin., *9*(04) (2022), 1-17.
[61] H. Zhang, F. Liu, I. Turner, and Q. Yang, *Numerical solution of the time fractional Black–Scholes model governing European options*, Comput. Math. Appl., *71*(9) (2016), 1772-1783.