A Cubic Spline Collocation Method to Solve a Nonlinear Space-Fractional Fisher’s Equation and Its Stability Examination

Adel R. Hadhoud 1, Faisal E. Abd Alaal 2, Ayman A. Abdelaziz 2 and Taha Radwan 3,4, *  

1 Department of Mathematics and Computer Science, Faculty of Science, Menoufia University, Shebin El-Kom 13829, Egypt  
2 Department of Mathematics, Faculty of Science, Damanhour University, Damanhour 22511, Egypt  
3 Department of Mathematics, College of Science and Arts in Ar-Rass, Qassim University, Ar Rass 51452, Saudi Arabia  
4 Department of Mathematics and Statistics, Faculty of Management Technology and Information Systems, Port Said University, Port Said 42511, Egypt  
* Correspondence: t.radwan@qu.edu.sa

Abstract: This article seeks to show a general framework of the cubic polynomial spline functions for developing a computational technique to solve the space-fractional Fisher’s equation. The presented approach is demonstrated to be conditionally stable using the von Neumann technique. A numerical illustration is given to demonstrate the proposed algorithm’s effectiveness. The novelty of the present work lies in the fact that the results suggest that the presented technique is accurate and convenient in solving such problems.

Keywords: Caputo sense; space-fractional Fisher’s equation; cubic polynomial spline; von Neumann stability

1. Introduction

Fractional calculus is a generalization of classical calculus that deals with fractional order differentiation and integration operations. Everyone who has studied elementary calculus is familiar with the differentiation operator, \( D = d/dx \). Additionally, the \( n \)th derivative of a suitable function, \( D^n f(x) = d^n f(x)/d^n x \), is properly defined as long as \( n \) is a positive integer. In 1695, L’Hôpital enquired to Leibniz about the meaning of \( D^n f \) if \( n \) was fractional. Many prominent mathematicians, including Euler, Laplace, Fourier, Abel, Liouville, Riemann, and Laurent, have since studied time-fractional calculus. However, it was not until 1884 that the theory of generalized operators progressed to the point where it could be used as a starting point for modern mathematicians. Fractional calculus has been utilized in thermodynamics, viscoelasticity, bioengineering, aerodynamics, control theory, electromagnetics, finance, chemistry, and signal processing in recent decades [1–3]. Fractional calculus theory dates back to Leibniz in the sixteenth century, and many different types of fractional operators have been created and developed virtually as far as the classical theory has evolved. The Riemann–Liouville, Caputo, conformable, and Riesz operator approaches are among these [4–9]. Fractional differential equations (FDEs) appear in engineering processes, a number of physical phenomena, financial products, and biological systems, such as non-exponential relaxation patterns and anomalous diffusion [10]. Several numerical methods have been developed to obtain an approximate solution of FDEs. These methods include the Adomian decomposition method [11], finite difference method [12], variational iteration method [13], spectral methods [14], and homotopy perturbation method [15,16].

Fisher [17] introduced the nonlinear fractional Fisher equation as a mathematical model to describe the kinetic advancing rate of an advantageous gene. The fractional
Fisher’s equation is also used in autocatalytic chemical reactions, nuclear reactor theory, neurophysiology, flame propagation, and branching Brownian motion processes [18–20]. In the open literature, many works describe solving the nonlinear fractional Fisher equation, see [20–23]. In our article, we propose a cubic polynomial spline-based technique for obtaining approximation solutions for the space-fractional Fisher’s equation in the following form [24]:

\[ \frac{\partial^\alpha u(x, t)}{\partial x^\alpha} = \frac{\partial u(x, t)}{\partial t} - u(x, t)(1 - u(x, t)) + f(x, t), x > 0, \]  

subject to the conditions:

\[ u(a, t) = \varphi_1(t), u(b, t) = \varphi_2(t), 0 < t \leq T, \]  

and

\[ u(x, 0) = g(x), a \leq x \leq b, \]  

where \( \alpha \) is a parameter in the Caputo sense that describes the fractional order of the space derivative, and \( 1 < \alpha \leq 2 \); if we take \( \alpha = 2 \), Equation (1) reduces to the classical nonlinear Fisher’s equation. Several scholars have studied and solved the space-fractional nonlinear Fisher’s equation using various numerical techniques, for example, the generalized differential transform method (GDTM) and the variational iteration method (VIM) [25], the radial basis functions method [26], the quadratic polynomial spline-based method (QPSM) [24], and the Legendre spectral collocation method [27]. The above-mentioned articles show how to solve the space-fractional Fisher’s equation in which the exact solution is unknown, but, as far as we are aware, no papers are using cubic polynomial spline for space-fractional Fisher’s equation; hence, that is the motivation for our study.

**Definition 1.** The Caputo space-fractional derivative operator of order \( \alpha > 0 \) is given by [25]:

\[
\mathcal{C}D^\alpha_x u(x, t) = \frac{\partial^\alpha u(x,t)}{\partial x^\alpha} = \begin{cases} 
\frac{1}{\Gamma(n-\alpha)} \int_{x}^{t} \frac{\partial^n u(\tau, t)}{\partial \tau^n} (x-\tau)^{n-\alpha-1} d\tau, & n-1 < \alpha < n, \\
\frac{\partial^n u(x,t)}{\partial x^n} \frac{\partial^\alpha u(x,t)}{\partial x^\alpha}, & \alpha = n \in \mathbb{N}, 
\end{cases}
\]

where \( \Gamma(\ ) \) is the gamma function. The right and left Caputo fractional derivatives of order \( \alpha \) are given, respectively, by [28]:

\[
\mathcal{C}D^\alpha_x u(x, t) = \frac{1}{\Gamma(n-\alpha)} \int_{x}^{b} (-1)^n \frac{\partial^n u(\tau, t)}{\partial \tau^n} (\tau-x)^{n-\alpha-1} d\tau, \ n-1 < \alpha \leq n, \\
\mathcal{C}D^\alpha_x u(x, t) = \frac{1}{\Gamma(n-\alpha)} \int_{a}^{x} \frac{\partial^n u(\tau, t)}{\partial \tau^n} (x-\tau)^{n-\alpha-1} d\tau, \ n-1 < \alpha \leq n.
\]

The numerical solutions of fractional differential equations using numerical methods with spline functions and their use is an important area of research in numerical analysis, see [29–34]. Such problems appear very often in physics, engineering, medicine, biology, and other sciences. Spline functions can be integrated and differentiated due to being piece-wise polynomials and, since they have a basis with small support, many of the integrals that occur in numerical methods are zero. Numerical methods with spline functions in obtaining the numerical solution of the differential equations lead to band matrices with a low computational cost. Cubic splines were used instead of using linear or quadratic splines, because they produce smooth approximations while maintaining a low computational cost.

**Definition 2.** Polynomial Spline Function [35,36]:

Let \( a = x_0 < \cdots < x_{n-1} < x_n = b \) be a subdivision of the interval \([a, b]\) and \( m \in \mathbb{N} \). A function \( P : [a, b] \to \mathbb{R} \) is called a spline of degree \( m \) for this subdivision if \( P \) is \( (m-1) \) times
continuously differentiable on \([a, b]\) and if the restriction of \(P\) to each subinterval \([x_i, x_{i+1}]\) for \(i = 0, 1, \cdots, n - 1\) reduces to a polynomial of degree at most \(m\). Thus, the polynomial spline function has the form:

\[
P(x) = \begin{cases} 
P_0(x), & \text{if } x \in [x_0, x_1], \\
P_1(x), & \text{if } x \in [x_1, x_2], \\
\vdots & \vdots \\
\vdots & \vdots \\
P_{n-1}(x), & \text{if } x \in [x_{n-1}, x_n]. 
\end{cases}
\]

such that each polynomial spline segment is given by:

\[
P_i(x) = \sum_{r=0}^{m} a_{ir}(x-x_r)^r, \quad i = 0, 1, \cdots, n-1, \quad x \in [x_i, x_{i+1}].
\]

The remainder of this article is structured as follows. In Section 2, a method is proposed that is based on the cubic spline polynomial method. In Section 3, the von Neumann method is used to discuss stability theoretically. In Section 4, we use a numerical example to demonstrate the suggested method’s efficiency and accuracy. The conclusion is presented in Section 5.

2. Derivation of the Method

To approximate \(u(x, t)\) by a cubic polynomial spline-based method over finite elements, let the region \(M = [a, b] \times [0, T]\) be discretized by a set of points \(M_{ij}\), which are the vertices of a grid of points \((x_i, t_j)\) where \(x_i = a + ih\), with \(h = \Delta x = \frac{b-a}{N}\) for each \(i = 0, 1, \cdots, N\) and \(t_j = jk, k = \Delta t\) for \(j = 0, 1, \cdots, R\).

Let \(W_i^j\) be an approximation to \(u(x_i, t_j)\) obtained by the segment \(P_i(x_i, t_j)\) of the spline function passing through the points \((x_i, W_i^j)\) and \((x_{i+1}, W_{i+1}^j)\). Each segment has the form [37, 38]:

\[
P_i(x, t_j) = a_1(t_j) (x-x_i)^3 + b_1(t_j) (x-x_i)^2 + c_1(t_j) (x-x_i) + d_1(t_j),
\]

for each \(i = 0, 1, \cdots, N-1\). To obtain expressions for the coefficients of Equation (4) in terms of \(W_i^j, W_{i+1}^j, S_i^j, \text{ and } S_{i+1}^j\), we first define:

\[
P_i(x_i, t_j) = W_i^j,
\]

(5)

\[
P_i(x_{i+1}, t_j) = W_{i+1}^j,
\]

(6)

\[
p_i^{(a)}(x_i, t_j) = \frac{\partial^a}{\partial x^a} P_i(x_i, t_j) = S_i^j,
\]

(7)

\[
p_i^{(a)}(x_{i+1}, t_j) = \frac{\partial^a}{\partial x^a} P_{i+1}(x_{i+1}, t_j) = S_{i+1}^j,
\]

(8)

where \(1 < \alpha \leq 2\). By using (4), (5), and (6), we have:

\[
d_i = W_i^j,
\]

(9)

\[
h^3a_i + h^2b_i + hc_i + d_i = W_{i+1}^j,
\]

(10)
where \( a_i \equiv a_i(t_i), b_i \equiv b_i(t_i), c_i \equiv c_i(t_i), \) and \( d_i \equiv d_i(t_i). \) Using the right Caputo fractional derivative and (4) in (7), we obtain:

\[
\frac{\partial^{\alpha}}{\partial x^{\alpha}} P_i(x, t) = \frac{1}{\Gamma(1-\alpha)} \int_{x_i}^{x_{i+1}} (\tau - x_i)^{1-\alpha} P_i^{(2)}(\tau, t_j) d\tau
\]
\[
= \frac{6(2-\alpha)h^{1-a}}{\Gamma(4-\alpha)} a_i + \frac{2h^{2-a}}{\Gamma(3-\alpha)} b_i = S_i^{(1)},
\]

and from using the left Caputo fractional derivative and (4) in (8), we obtain:

\[
\frac{\partial^{\alpha}}{\partial t^{\alpha}} P_i(x, t) = \frac{1}{\Gamma(1-\alpha)} \int_{x_i}^{x_{i+1}} (x_i - \tau)^{1-\alpha} P_i^{(2)}(\tau, t_j) d\tau
\]
\[
= \frac{6h^{1-a}}{\Gamma(4-\alpha)} a_i + \frac{2h^{1-a}}{\Gamma(5-\alpha)} b_i = S_i^{(1)},
\]

We obtain the following expressions by solving (9)–(12):

\[
a_i = \frac{\Gamma(4-\alpha)}{6(1-\alpha)h^{1-a}} (S_i^{(1)} - S_i^{(1+1)}),
\]
\[
b_i = \frac{\Gamma(3-\alpha)}{2(1-\alpha)h^{2-a}} (2 - \alpha) S_i^{(1)} - S_i^{(1)},
\]
\[
c_i = \frac{1}{h} (W_i^{(1)} - W_i^{(1-1)}) + \frac{\Gamma(4-\alpha)}{6(1-\alpha)h^{1-a}} (S_i^{(1)} - S_i^{(1+1)}) + \frac{\Gamma(3-\alpha)}{2(1-\alpha)h^{1-a}} (S_i^{(1)} - (2 - \alpha) S_i^{(1+1)}),
\]
\[
d_i = W_i^{(1)}.
\]

Using the first derivative’s continuity condition at \( x = x_i, \) that is \( P_i^{(1)}(x_i, t_j) = P_i^{(1)}(x_i, t_j), \) we obtain:

\[
c_i - c_i^{(1-1)} = 3h^2 a_i^{(1)} + 2hb_i^{(1)}
\]

Substituting the expressions \( a_i, b_i, \) and \( c_i \) from Equation (13) into Equation (14), we obtain:

\[
W_i^{(1)} - 2W_i^{(1-1)} + W_i^{(1-2)} + \frac{\Gamma(4-\alpha)}{6(1-\alpha)h^{1-a}} (S_i^{(1)} - S_i^{(1+1)}) + \frac{\Gamma(3-\alpha)}{2(1-\alpha)h^{1-a}} (S_i^{(1)} - (2 - \alpha) S_i^{(1+1)})
\]
\[
= \frac{\Gamma(4-\alpha)}{6(1-\alpha)h^{1-a}} (S_i^{(1-1)} - S_i^{(1-2)}) + \frac{\Gamma(3-\alpha)}{2(1-\alpha)h^{1-a}} (S_i^{(1)} - (2 - \alpha) S_i^{(1+1)}).
\]

Equation (15) can be rewritten in the form:

\[
= \left( \frac{\Gamma(4-\alpha)}{2(1-\alpha)h^{1-a}} - \frac{\Gamma(3-\alpha)}{6(1-\alpha)h^{1-a}} \right) S_i^{(1-1)}
\]
\[
+ \left( \frac{2\Gamma(4-\alpha)}{6(1-\alpha)h^{1-a}} - \frac{\Gamma(3-\alpha)}{2(1-\alpha)h^{1-a}} \right) S_i^{(1)}
\]
\[
- \left( \frac{2\Gamma(4-\alpha)}{6(1-\alpha)h^{1-a}} - \frac{\Gamma(3-\alpha)}{2(1-\alpha)h^{1-a}} \right) S_i^{(1+1)}
\]
\[
+ \left( \frac{\Gamma(3-\alpha)}{2(1-\alpha)h^{1-a}} - \frac{\Gamma(4-\alpha)}{6(1-\alpha)h^{1-a}} \right) S_i^{(1+1)}
\]

After slight rearrangements, Equation (16) becomes:

\[
W_i^{(1)} - 2W_i^{(1-1)} + W_i^{(1-2)} = \beta S_i^{(1-1)} + \gamma S_i^{(1)} + \beta S_i^{(1+1)}, i = 1, \ldots, N - 1,
\]

where \( \beta = \frac{(2\alpha-3)h^2 \Gamma(3-\alpha)}{6(1-\alpha)} \) and \( \gamma = \frac{4h^2 \Gamma(3-\alpha)}{6(1-\alpha)} \). As \( \alpha \to 2, \) system (17) reduces to:

\[
W_i^{(1)} - 2W_i^{(1-1)} + W_i^{(1-2)} = \frac{1}{6} (S_i^{(1-1)} + 4S_i^{(1)} + S_i^{(1+1)}), i = 1, \ldots, N - 1.
\]
Remark 1: For Equation (17), the local truncation error is:

\[ T_i^{ij} = \beta(u_{i-1} + u_{i+1}) - 2\gamma u_i - \beta\left(D_x^2 u_{i-1} + D_x^2 u_{i+1}\right) - \gamma D_x^2 u_i \]  

(19)

and can be obtained by expanding this equation in terms of \( u(x_i, t_f) \) and its derivatives in the Taylor series, as follows:

\[ T_i^{ij} = \left(h^2 - (\gamma + 2\beta)\right)D_x^2 u_i + \left(h^4 - \beta h^2\right)D_x^4 u_i + \left(h^6 - \frac{\beta h^4}{12}\right)D_x^6 u_i + \ldots \]  

(20)

where \( \theta = \Gamma(3 - \alpha) \) and \( \delta = \frac{(2\alpha - 3)\Gamma(3 - \alpha)}{6\alpha - 1} \). From the expression (21) of the local truncation error, our scheme (17) is reduced to \( O(h^a) \), \( 1 < a \leq 2 \).

The fractional Fisher’s Equation (1) can now be defined as follows in terms of \( S_i^j \):

\[ S_i^j = \frac{\partial^a W_i^j}{\partial x^a} = \frac{\partial W_i^j}{\partial t} - W_i^j \left(1 - W_i^j\right) + f(x_i, t_f) \]  

Using the finite difference method, we obtain:

\[ \frac{\partial W_i^j}{\partial t} \approx \frac{W_i^j - W_i^{j-1}}{k} \]  

This allows us to express \( S_{i-1}^j, S_i^j, \) and \( S_{i+1}^j \) as follows:

\[ S_{i-1}^j = \frac{W_{i+1}^j - W_i^{j-1}}{k} - \sigma_i^1 W_{i-1}^j + f_i^1 \]  
\[ S_i^j = \frac{W_i^j - W_{i-1}^j}{k} - \sigma_i^1 W_i^j + f_i^1 \]  
\[ S_{i+1}^j = \frac{W_i^j - W_{i+1}^j}{k} - \sigma_i^1 W_{i+1}^j + f_{i+1}^1 \]  

(22)

where \( \sigma_i^1 = \left(1 - W_i^j\right) \). Substituting (22) into (17) gives us the following system:

\[ A_i W_{i-1}^j + B_i W_i^j + C_i W_{i+1}^j = A_i^* W_{i-1}^{j-1} + B_i^* W_i^{j-1} + C_i^* W_{i+1}^{j-1} + \rho_i^j, \]

\[ i = 1, \ldots, N - 1; \quad j = 1, \ldots, R, \]  

(23)

where:

\[ A_i = k - \beta + \beta k \sigma_{i-1}^j, \quad A_i^* = -\beta, \]  
\[ B_i = -2k - \gamma + \gamma k \sigma_i^j, \quad B_i^* = -\gamma, \]  
\[ C_i = k - \beta + \beta k \sigma_{i+1}^j, \quad C_i^* = -\beta, \]

and:

\[ \rho_i^j = \beta k f_{i-1}^j + \gamma k f_i^j + \beta k f_{i+1}^j. \]

System (23) consists of \( N - 1 \) equations in the unknowns \( W_i^j, i = 0, 1, \ldots, N \). Two additional equations are required to solve this system. The boundary conditions in (2) are used to derive these equations, which can be represented as:

\[ W_0^j = \varphi_1(t_j), \quad W_N^j = \varphi_2(t_j), \quad j = 0, 1, \ldots, R. \]  

(24)
Writing (23) and (24) in matrix form gives:
\[
\Phi W^j = \Phi^* W^{j-1} + r^j, \tag{25}
\]
where:
\[
W^j = \begin{pmatrix} W^j_0, W^j_1, W^j_2, \ldots, \ldots, W^j_{N-1}, W^j_N \end{pmatrix}^T
\]
\[
\Phi = \begin{pmatrix}
1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
A_1 & B_1 & C_1 & 0 & \cdots & 0 & 0 & 0 \\
0 & A_2 & B_2 & C_2 & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \cdots & A_{N-1} & B_{N-1} & C_{N-1} \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1
\end{pmatrix}
\]
\[
\Phi^* = \begin{pmatrix}
0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
A^*_1 & B^*_1 & C^*_1 & 0 & \cdots & 0 & 0 & 0 \\
0 & A^*_2 & B^*_2 & C^*_2 & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \cdots & A^*_{N-1} & B^*_{N-1} & C^*_{N-1} \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0
\end{pmatrix}
\]
and,
\[
r^j = \begin{pmatrix} \varnothing_1(t_j), \varrho^j_1, \varrho^j_2, \ldots, \ldots, \varrho^j_{N-1}, \varnothing_2(t_j) \end{pmatrix}^T
\]
where \(\Phi^*\) and \(\Phi\) are \((N + 1) \times (N + 1)\) matrices and \(W^j\) and \(r^j\) are \((N + 1)\) vectors.

Now, the initial condition, \(u(x, t_0) = g(x)\) for \(a \leq x \leq b\), implies that \(W^0_i\) for \(i = 0, 1, \ldots, N\). These values can be utilized in (25) to obtain the value of \(W^1_i\) for \(i = 0, 1, \ldots, N\). If the technique is reapplied once all the approximations \(W^1_i\) are determined, the values of \(W^2_i, W^3, \ldots, W^K_i\) can be derived in the same way.

The nonlinear term in the system is linearized using the technique in (25). For example, if \(j = 1\), we approximate \(\sigma^1_i\) by \(\sigma^1_i^*\) computed from \(w^0_i\), which obtains an approximation to \(w^1_i\). For \(j = m\), we approximate \(\sigma^m_i\) by \(\sigma^m_i^*\) computed from \(w^{m-1}_i\) and obtain an approximation to \(w^m_i\).

### 3. Stability Analysis

The von Neumann technique will be used to investigate our system’s stability (23). To do this, we must linearize the nonlinear term \(u(x, t)\left(1 - u(x, t)\right)\) of Fisher’s Equation (1), by assuming that the corresponding quantities \(\sigma^j_{i-1}, \sigma^j_i,\) and \(\sigma^j_{i+1}\) are equal to a local constant \(\lambda^*\) in (23). The key part of the von Neumann analysis is to assume a solution of the form [39]:
\[
W^j_i = \xi^j \exp(\varphi q_i h),
\]
where \(q = \sqrt{-1}\), \(h\) is the element size, \(\varphi\) is the mode number, and \(\xi^j\) is the amplification factor at time level \(j\). As \(j\) increases (more time steps are computed), adding the latter \(W^j_i\) expression to the system (23) gives the following characteristic equation:
\[
\xi^j (A_i \exp(\varphi q(i-1)h) + B_i \exp(\varphi qih) + C_i \exp(\varphi q(i+1)h)) = \xi^{j-1} (A^*_i \exp(\varphi q(i-1)h) + B^*_i \exp(\varphi qih) + C^*_i \exp(\varphi q(i+1)h)),
\]
where:
\[
A_i = k - \beta + \beta k \lambda^*, \quad A^*_i = -\beta,
\]
\[
B_i = -2k - \gamma + \gamma k \lambda^*, \quad B^*_i = -\gamma,
\]
\[
C_i = k - \beta + \beta k \lambda^*, \quad C^*_i = -\beta,
\]
After simple basic calculations, (27) becomes:

\[
\xi = \frac{A_i^* \exp(-q\phi) + B_i^* + C_i^* \exp(q\phi)}{A_i \exp(-q\phi) + B_i + C_i \exp(q\phi)},
\]

where \(\phi = qh\). After using Euler’s formula \(\exp(q\phi) = \cos \phi + q \sin \phi\), the last equation (28) can be simplified as:

\[
\xi = \frac{A_i^* + C_i^*}{A_i + C_i} \cos \phi + B_i^* + q(C_i^* - A_i^*) \sin \phi.
\]

Equation (29) can be rewritten in the forms:

\[
\xi = \frac{-2\beta \cos \phi - \gamma}{(2k - 2\beta + 2\beta k \lambda^*) \cos \phi + (-2k - \gamma + \gamma k \lambda^*)}.
\]

or

\[
\xi = \frac{(2\beta \cos \phi + \gamma)}{(2\beta \cos \phi + \gamma) + 2k(1 - \cos \phi - \lambda^*(2\beta k \cos \phi + \gamma k)}.
\]

The quantity \((1 - \cos \phi)\) is positive or equal to zero but \(2\beta \cos \phi + \gamma\) is surely positive if we choose \(\gamma > 0\) and \(\beta > 0\) such that \(\gamma > 2\beta\). If we choose \(k\), \(\beta\), and \(\gamma\) small enough to make \(\lambda^*(2\beta k \cos \phi + \gamma k) \rightarrow 0\), then last (31) is close to:

\[
\xi = \frac{(2\beta \cos \phi + \gamma)}{(2\beta \cos \phi + \gamma) + 2k(1 - \cos \phi)}.
\]

For stability, we must have \(|\xi| \leq 1\) (otherwise \(\xi\) in (26) would expand in an unbounded manner). This condition is satisfied for \(\gamma > 0\) and \(\beta > 0\) such that \(\gamma > 2\beta\). Finally, we can say that our numerical scheme is conditionally stable for \(\gamma > 0\), \(\beta > 0\), and \(\gamma > 2\beta\), if \(k\), \(\beta\), and \(\gamma\) values chosen are small enough.

4. Numerical Results

In this part, we employ the suggested method for solving the nonlinear space-fractional Fisher’s equation [17,18] with \(\alpha = 1.5\), \(f(x,t) = -x^2\), and \(u(x,0) = x\) and subject to conditions:

\[
u(0.0125, t) \approx 0.0125(1 + t) + 0.00609375 l^2 - 0.082176 l^3 - 0.0210541 l^4 - 7.16634 \times 10^{-6} l^5,
\]

\[
u(1.0125, t) \approx 1.0125(1 + t) - 0.518906 l^2 - 0.921366 l^3 + 0.310529 l^4 + 0.0845434 l^5.
\]

The solution of the space-fractional Fisher’s equation, like that of most fractional partial differential equations, is unknown, so we will compare it with the known computational approaches such as GDTM and VIM. We used \(k = 0.0005\) and \(N = 80\) in all our numerical results. Tables 1–3 compare our method, which was improved in Section 2, to other existing methods.

Table 1. The comparison between the proposed method and other existing methods for \(t = 0.1\) and various \(x\) values.

| \(x\) | Present Method | VIM [18] | GDTM [18] | QPSM [17] |
|------|----------------|----------|-----------|-----------|
| 0.0  | 0.000000       | 0.000000 | 0.000000  | 0.000000  |
| 0.2  | 0.220600       | 0.220589 | 0.220348  | 0.210917  |
| 0.4  | 0.440452       | 0.440329 | 0.439957  | 0.425837  |
| 0.6  | 0.659483       | 0.659214 | 0.658707  | 0.645538  |
| 0.8  | 0.877676       | 0.877185 | 0.876385  | 0.869671  |
| 1.0  | 1.094273       | 1.094096 | 1.093387  | 1.093920  |
Table 2. The comparison between the proposed method and other existing methods for \( t = 0.2 \) and various \( x \) values.

| \( x \) | Present Method | VIM [18] | GDTM [18] | QPSM [17] |
|-------|----------------|---------|-----------|-----------|
| 0.0   | 0.000000       | 0.000000| 0.000000  | 0.000000  |
| 0.2   | 0.241796       | 0.242240| 0.240212  | 0.223360  |
| 0.4   | 0.480839       | 0.480928| 0.477796  | 0.453229  |
| 0.6   | 0.716555       | 0.716018| 0.711692  | 0.691755  |
| 0.8   | 0.948791       | 0.947058| 0.941796  | 0.936764  |
| 1.0   | 1.173899       | 1.172904| 1.168067  | 1.173100  |

Table 3. The comparison between the proposed method and other existing methods for \( t = 0.4 \) and various \( x \) values.

| \( x \) | Present Method | VIM [18] | GDTM [18] | QPSM [17] |
|-------|----------------|---------|-----------|-----------|
| 0.0   | 0.000000       | 0.000000| 0.000000  | 0.000000  |
| 0.2   | 0.283243       | 0.287146| 0.269327  | 0.275929  |
| 0.4   | 0.557963       | 0.559433| 0.531749  | 0.542191  |
| 0.6   | 0.819056       | 0.816392| 0.777297  | 0.798553  |
| 0.8   | 1.064467       | 1.054714| 1.004978  | 1.044459  |
| 1.0   | 1.273878       | 1.265508| 1.214400  | 1.277563  |

Tables 1–3 show that for all values of \( x \) and \( t \), our approximate solutions are in good agreement with the approximate solutions using other existing methods. This is also shown in Figures 1 and 2.

Figure 1. The behaviors of the numerical solution \( W(x, t) \) for various time levels.
Table 3. The comparison between the proposed method and other existing methods for $t = 0.4$ and various $x$ values.

| $x$ | Present Method | VIM [18] | GDTM [18] | QPSM [17] |
|-----|----------------|----------|-----------|-----------|
| 0.0 | 0.000000       | 0.000000 | 0.000000  | 0.000000  |
| 0.2 | 0.283243       | 0.287146 | 0.269327  | 0.275929  |
| 0.4 | 0.557963       | 0.559433 | 0.531749  | 0.542191  |
| 0.6 | 0.819056       | 0.816392 | 0.777297  | 0.798553  |
| 0.8 | 1.064467       | 1.054714 | 1.004978  | 1.044459  |
| 1.0 | 1.273878       | 1.265508 | 1.214400  | 1.277563  |

5. Conclusions

The nonlinear space-fractional Fisher's equation is numerically treated in this article using a collocation method with cubic polynomial spline functions. The proposed method is demonstrated to be conditionally stable using the von Neumann stability technique. Compared to other useful techniques, such as VIM, GDTM, and QPSM, the numerical results illustrate that our proposed method retains efficiency, capability, and good versatility. The findings of our study are hoped to pique the attention of academics interested in using the cubic polynomial spline to numerically solve nonlinear fractional partial differential equations of the same type. It is proven that our scheme's local truncation error is to $O(h^\alpha)$, $1 < \alpha \leq 2$. Furthermore, the novelty of the present work lies in the fact that the results illustrate that the proposed technique is convenient, accurate, and very efficient in solving such problems.

Author Contributions: A.R.H., F.E.A.A., A.A.A. and T.R.; methodology, A.R.H., F.E.A.A., A.A.A. and T.R.; software, A.R.H., F.E.A.A., A.A.A. and T.R.; validation, A.R.H., F.E.A.A., A.A.A. and T.R.; formal analysis, A.R.H., F.E.A.A., A.A.A. and T.R.; investigation, A.R.H., F.E.A.A., A.A.A. and T.R.; resources, A.R.H., F.E.A.A., A.A.A. and T.R.; data curation, A.R.H., F.E.A.A., A.A.A. and T.R.; writing—original draft preparation, A.R.H., F.E.A.A., A.A.A. and T.R.; writing—review and editing, A.R.H., F.E.A.A., A.A.A. and T.R.; visualization, A.R.H., F.E.A.A., A.A.A. and T.R.; supervision, A.R.H.; project administration, T.R.; funding acquisition, All authors have read and agreed to the published version of the manuscript.

Funding: This research has received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: The researchers would like to thank the Deanship of Scientific Research, Qassim University, for funding the publication of this project.

Conflicts of Interest: All the authors declare that they have no conflict of interest.
33. Akram, T.; Abbas, M.; Ismail, A.I.; Ali, N.H.M.; Baleanu, D. Extended cubic B-splines in the numerical solution of time fractional telegraph equation. *Adv. Differ. Equ.* 2019, 2019, 1–20. [CrossRef]

34. Madiha, S.; Muhammad, A.; Farah, A.A.; Abdul, M.; Thabet, A.; Manar, A.A. Numerical solutions of time fractional Burgers’ equation involving Atangana–Baleanu derivative via cubic B-spline functions. *Results Phys.* 2022, 34, 105244.

35. Schumaker, L. *Spline Functions, Basic Theory*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 1981.

36. Ahlberg, J.H.; Nilson, E.N.; Walsh, J.L. The Theory of Splines and their Applications. In *Mathematics in Science and Engineering*; Academic Press: Cambridge, MA, USA, 1967; Available online: https://www.sciencedirect.com/bookseries/mathematics-in-science-and-engineering/vol/38/suppl/C (accessed on 19 July 2022).

37. El-Danaf, T.S.; Ramadan, M.A.; Abd Alaal, F.E.I. Numerical studies of the cubic non-linear Schrodinger equation. *Nonlinear Dyn.* 2012, 67, 619–627. [CrossRef]

38. Ramadan, M.A.; Lashien, I.F.; Zahra, W.K. Polynomial and nonpolynomial spline approaches to the numerical solution of second order boundary value problems. *Appl. Math. Comput.* 2007, 184, 476–484. [CrossRef]

39. Ramadan, M.A.; El-Danaf, T.S.; Abd Alaal, F.E.I. A numerical solution of the Burgers’ equation using septic B-splines. *Chaos Solitons Fractals* 2005, 26, 1249–1258. [CrossRef]