Research Article

A B-Spline Quasi Interpolation Crank–Nicolson Scheme for Solving the Coupled Burgers Equations with the Caputo–Fabrizio Derivative

M. Taghipour1 and H. Aminikhah

1Department of Applied Mathematics and Computer Science, Faculty of Mathematical Sciences, University of Guilan, P. O. Box 1914, Rasht 41938, Iran
2Center of Excellence for Mathematical Modelling, Optimization and Combinational Computing (MMOCC), University of Guilan, P. O. Box 1914, Rasht 41938, Iran

Correspondence should be addressed to H. Aminikhah; aminikhah@guilan.ac.ir

Received 8 September 2020; Revised 29 December 2020; Accepted 8 January 2021; Published 23 January 2021

Academic Editor: Li-Tao Zhang

Copyright © 2021 M. Taghipour and H. Aminikhah. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, a Crank–Nicolson finite difference scheme based on cubic B-spline quasi-interpolation has been derived for the solution of the coupled Burgers equations with the Caputo–Fabrizio derivative. The first- and second-order spatial derivatives have been approximated by first and second derivatives of the cubic B-spline quasi-interpolation. The discrete scheme obtained in this way constitutes a system of algebraic equations associated with a bi-pentadiagonal matrix. We show that the proposed scheme is unconditionally stable. Numerical examples are provided to verify the efficiency of the method.

1. Introduction

The coupled Burgers equations are coupled partial differential equations which are capable of describing realistic polydisperse suspensions. The coupled Burgers equation predicts an interesting phenomenon, which is called phase shifts [1]. This equation is one of the fundamental models in fluid mechanics and arises in gas dynamics, chromatography, and flood waves in rivers [2]. The coupled viscous Burgers equation is given by

\begin{align*}
    u_t - u_{xx} + \eta uu_x + \alpha_1 (uv)_x &= 0, \quad x \in [a, b], \ t \in [0, T],
    
    v_t - v_{xx} + \eta vv_x + \alpha_2 (uv)_x &= 0, \quad x \in [a, b], \ t \in [0, T],
\end{align*}

with the initial conditions

\begin{align*}
    u(x, 0) &= \phi_1 (x), \\
    v(x, 0) &= \phi_2 (x),
\end{align*}

and the boundary conditions

\begin{align*}
    u(a, t) &= f_1 (a, t), \\
    u(b, t) &= f_2 (b, t), \\
    v(a, t) &= g_1 (a, t), \\
    v(b, t) &= g_2 (b, t),
\end{align*}

where \( \eta \) is a real constant and \( \alpha_1 \) and \( \alpha_2 \) are arbitrary constants depending on the system parameters such as Peclet number, Stokes velocity of particles due to gravity, and Brownian diffusivity [3].

Spline is a special function defined piecewise by polynomials. The spline approximation first appeared in a paper by Schoenberg [4]. Spline interpolation is a form of interpolation where the interpolant is a special type of piecewise polynomial called a spline. Applications of spline function in fractional partial differential equations can be found in [5–15].
In this article, we consider the following coupled Burgers equation with time fractional derivative:

\[
\begin{align*}
\frac{\partial^\alpha u}{\partial t^\alpha} - \frac{\partial^\beta u}{\partial x^\beta} + \eta \frac{\partial u}{\partial x} + \alpha_1 \frac{\partial (uv)}{\partial x} &= q_1, \\
\frac{\partial^\gamma v}{\partial t^\gamma} - \frac{\partial^\delta v}{\partial x^\delta} + \eta \frac{\partial v}{\partial x} + \alpha_2 \frac{\partial (uv)}{\partial x} &= q_2,
\end{align*}
\]

with initial

\[
\begin{align*}
u(x, 0) &= \phi_1(x), \\
v(x, 0) &= \phi_2(x),
\end{align*}
\]

and the boundary conditions

\[
\begin{align*}
u(a, t) &= f_1(a, t), \\
u(b, t) &= f_2(b, t), \\
\eta \frac{\partial v}{\partial x}(a, t) &= g_1(a, t), \\
\eta \frac{\partial v}{\partial x}(b, t) &= g_2(b, t),
\end{align*}
\]

where \( \gamma \) is order of time fractional derivative. Also, \( \eta, \alpha_1 \), and \( \alpha_2 \) are those ones we said before.

\[
\frac{\partial^\alpha u}{\partial t^\alpha} = ((\partial^\alpha u(x, t))/\partial t^\alpha) \quad \text{denotes the Caputo–Fabrizio derivative of the function } u(x, t) \text{ defined as [16]}
\]

\[
\frac{\partial^\alpha u}{\partial t^\alpha} = \frac{M(\gamma)}{1 - \gamma} \int_0^t u'(x, s)e^{-\sigma(t-s)}ds,
\]

where \( M(\gamma) \) is a normalization function such that \( M(0) = M(1) = 1 \) and \( \sigma = (\gamma/1 - \gamma) \).

Recently, the Caputo–Fabrizio derivative has received more attention from researchers due to their description of some physical phenomenon [17–26].

For the description of memory and some physical properties of various materials and processes, modeling with fractional derivatives is very appropriate. This is the main benefit of fractional derivatives in comparison with classical integer order models, in which such effects are missed. In recent years, the coupled system of Burgers equations with fractional derivatives has been the focus of attention. For example, in [27], the Adomian decomposition method is directly extended to study the coupled Burgers equations with time and space fractional derivatives. Khan et al. [28] proposed the generalized differential transform method (GDTM) and homotopy perturbation method (HPM) for time fractional Burgers and coupled Burgers equations. The fractional variational iteration method (FVIM) to solve a time and space fractional coupled Burgers equations is given by Prakash et al. [29]. In [30], a q-homotopy analysis transform method (q-HATM) for time and space fractional coupled Burgers equations is introduced. Aminikhah and Malekzadeh [31] introduced a new homotopy perturbation method for system of variable coefficient coupled Burgers equations with time fractional derivative. In [32], the Laplace-Adomian decomposition method (LADM), the Laplace-variational iteration method (LVIM), and the reduced differential transform method (RDTM) are proposed to solve the one- and two-dimensional fractional coupled Burgers equations. Albuohimid and Adibi derived a hybrid spectral exponential Chebyshev method (HSECM) for time fractional coupled Burgers equations [33]. In [34], authors investigate the fractional coupled viscous Burgers equations involving Mittag–Leffler kernel. In [35], the generalized two-dimensional differential transform method (DTM) was applied to solve the coupled Burgers equations with space and time fractional derivatives. Ozbudemir et al. used the Gegenbauer wavelets-based computational method to find the approximate solutions of the coupled system of Burgers equations with time fractional derivative [36].

Our aim is to propose a Crank–Nicolson finite difference scheme using cubic B-spline quasi-interpolation to solve time fractional coupled viscous Burgers equations. The first- and second-order spatial derivatives have been approximated by first and second derivatives of the cubic B-spline quasi-interpolation. This approximations have not been used for the fractional coupled Burgers equations before.

The paper is organized as follows. In Section 2, we present some basic definitions and concepts of quasi-interpolants. In Section 3, using the quasi-interpolant and Crank–Nicolson finite difference method, we obtain a numerical scheme. The stability of this method is studied in Section 4. In Section 5, some numerical examples are proposed. Finally, conclusions are given in Section 6.

### 2. Univariate Spline Quasi-Interpolants

In this section, we introduce the basic concepts about B-spline and univariate B-spline quasi-interpolants that we will use in Section 3.

According to [37], let

\[
P_d := \text{space of univariate polynomials of degree at most } d,
\]

and \( \Omega = [a, b] \) be an interval that has been partitioned into subintervals via a set of points \( \Delta = \{x_i\}_{i=0}^k \) with

\[
a = x_0 < x_1 < \ldots < x_k < x_{k+1} = b.
\]

We define the space of univariate polynomial splines of smoothness \( r \) and degree \( d \) with knots \( \Delta \) as

\[
\delta_d^r(\Delta) = \left\{ s \in C^r(\Omega) : s |_{(x_i, x_{i+1})} \in P_d, \quad i = 0, \ldots, k \right\},
\]

where \( 0 \leq r < d \) are given integers. We have

\[
n := \text{dim } \delta_d^r(\Delta) = k(d - r) + r + 1.
\]

For a formal proof of this fact, see Theorem 4.4 of [38].

Given \( 0 \leq r < d \) and \( \Delta = \{x_i\}_{i=0}^{k+1} \), the associated extended partition \( \Delta_x \) is defined to be \( \{y_j\}_{j=0}^{n+d+1} \), where \( n \) is the dimension of \( \delta_d^r(\Delta) \) given in (11):
\[
a = y_1 = \cdots = y_{d+1},
\]
\[
y_{m+1} = \cdots = y_{md+1} = b,
\]
\[
y_{d+2} \leq \cdots \leq y_n = x_1, \ldots, x_k.
\]

Given an extended partition \( \Delta_n \), let
\[
Q_i^1(t) = \begin{cases} 
\frac{1}{y_{i+1} - y_i}, & y_i \leq t < y_{i+1}, \\
0, & \text{otherwise,}
\end{cases}
\]
for \( i = 1, \ldots, n + d \), and let
\[
Q_i^m(t) = \begin{cases} 
\frac{(t - y_i)Q_{i+1}^{m-1}(t) + (y_{i+m} - t)Q_{i-1}^{m-1}(t)}{y_{i+m} - y_i}, & y_i \leq t < y_{i+m}, \\
0, & \text{otherwise,}
\end{cases}
\]
for \( 2 \leq m \leq d + 1 \) and \( i = 1, \ldots, n + d - m + 1 \). Let
\[
N_i^m(t) := (y_{i+m} - y_i)Q_i^m(t), \quad i = 1, \ldots, n + d - m + 1.
\]

We call these the normalized B-splines of order \( m \) (or degree \( m - 1 \)) associated with the extended partition \( \Delta_n \).

In [37], univariate B-spline quasi-interpolants can be defined as a formula of the form
\[
Q_d f(x) = \sum_{i=1}^{md} (\lambda_i, f) N_i^m(x),
\]
where \( \{N_{i,j}^{md}\} \) are the B-splines forming a basis of \( \delta_d^m(\Delta) \).

Quasi-interpolants have been heavily studied in the literature. Some basic ideas and sources for further information can be found in [38]. For a good approximations, we need to make sure it reproduces polynomials, i.e., \( Qp = p \) for all \( p \in P_d \). For each \( i = 1, \ldots, n \), we assume that the coefficient \( \lambda_i \) is a linear functional defined on \( C[a, b] \) that can be computed from samples of \( f \) at some set of points \( \sigma(\lambda_i) \) in \([a, b]\).

According to [39], the error of a quasi-interpolation satisfies
\[
|f(x) - (Q_d f)(x)| \leq \frac{\|Q_d\|_{d+1}}{(d+1)!} \|f^{(d+1)}\|_{\infty, D_d} h(x)^{d+1}, \quad x \in D_d,
\]
where \( D_d = [y_{d+1}, y_{m+1}] \), \( D_d \) is the union of the supports of all B-splines \( N_i, i \sim x \) and \( \|f^{(d+1)}\|_{\infty, D_d} \) denote the maximum norm of \( f^{(d+1)} \) on \( D_d \) and \( h(x) = \max_{y \in D_d} |y - x| \) that is used to indicate proportionality. If the local mesh ratio is bounded, i.e., if the quotients of the lengths of adjacent knot intervals are \( \leq r \), then the error of the derivatives on the knot intervals \( (y_i, y_{i+1}) \) can be estimated by
\[
|f^{(j)}(x) - (Q_d f)^{(j)}(x)| \leq c(d, r) \|Q_d\|_{d+1} \|f^{(d+1)}\|_{\infty, D_d} h(x)^{d+1-j},
\]
for \( j < d \).

Suppose \( a = t_0 < \cdots < t_n = b \) are equally spaced points in the interval \([a, b]\). Let
\[
\lambda_i f := \begin{cases} 
(f(t_0)), & i = 1, \\
\frac{1}{18} (7 f(t_0) + 18 f(t_1) - 9 f(t_2) + 2 f(t_3)), & i = 2, \\
\frac{1}{6} (-f(t_{i-3}) + 8 f(t_{i-2}) - f(t_{i-1})), & 3 \leq i \leq n + 1, \\
\frac{1}{18} (2 f(t_{n-3}) - 9 f(t_{n-2}) + 18 f(t_{n-1}) + 7 f(t_n)), & i = n + 2, \\
f(t_n), & i = n + 3.
\end{cases}
\]
\[ f'_i = \sum_{i=1}^{n+3} (\lambda_i f) N'_i (x), \quad i = 0, 1, \ldots, n, \]
\[ f''_i = \sum_{i=1}^{n+3} (\lambda_i f) N''_i (x), \quad i = 0, 1, \ldots, n, \]

we obtain
\[
Y' = \frac{1}{h}D^1 Y, \\
Y'' = \frac{1}{h^2}D^2 Y,
\]

where \( D^1, D^2 \in \mathbb{R}^{(n+1) \times (n+1)} \) and is obtained as follows:

\[
D^1 = \begin{pmatrix}
-\frac{11}{6} & 3 & -\frac{3}{2} & 1 & 0 & 0 & \cdots & 0 & 0 \\
-\frac{1}{3} & -\frac{1}{2} & 1 & -\frac{1}{6} & 0 & 0 & \cdots & 0 & 0 \\
\frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} & -\frac{1}{12} & 0 & \cdots & 0 & 0 \\
0 & \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} & -\frac{1}{12} & \cdots & 0 & 0 \\
0 & 0 & \cdots & \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} & -\frac{1}{12} & 0 \\
0 & 0 & \cdots & 0 & \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} & -\frac{1}{12} \\
0 & 0 & \cdots & 0 & 0 & \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} \\
0 & 0 & \cdots & 0 & 0 & \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} \\
0 & 0 & \cdots & 0 & 0 & \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} \\
0 & 0 & \cdots & 0 & 0 & \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} \\
\end{pmatrix}
\]

\[
D^2 = \begin{pmatrix}
2 & -5 & 4 & -1 & 0 & 0 & \cdots & 0 & 0 \\
1 & -2 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\
-\frac{1}{6} & 5 & -3 & 5 & \frac{1}{6} & 0 & \cdots & 0 & 0 \\
0 & -\frac{1}{6} & 5 & -3 & 5 & \frac{1}{6} & 0 & \cdots & 0 & 0 \\
0 & 0 & \cdots & -\frac{1}{6} & 5 & -3 & 5 & \frac{1}{6} & 0 \\
0 & 0 & \cdots & 0 & -\frac{1}{6} & 5 & -3 & 5 & \frac{1}{6} \\
0 & 0 & \cdots & 0 & 0 & -1 & 4 & -5 & 2 \\
\end{pmatrix}
\]

3. Numerical Scheme

We consider a grid \( x_i = a + ih, i = 0, 1, \ldots, M, \) with \( h = x_{i+1} - x_i. \) The step length in time is denoted by \( \tau \) and \( t_k = k\tau, \tau = (T/N), 0 \leq k \leq N. \) So, the domain \( [a, b] \times [0, T] \) is divided into a uniform grid of mesh points \((x_j, t_k)\). The values of the function \( u \) at the grid points are denoted \( u^k_j = u(x_j, t_k) \) and \( U^k_j \) is the approximate solution at the point \((x_j, t_k)\).

A discrete approximation to the \( \frac{\partial}{\partial t} D^2 u(x, t) \) at \((x_j, t_k)\) can be obtained by the following approximation [40]:

\[
\frac{\partial}{\partial t} D^2 u(x_j, t_k) = \frac{M(y)}{1 - y} \int_0^t \frac{\partial u(x_j, s)}{\partial s} e^{-\frac{y}{\tau} \sigma(t_k - s)} ds \\
= \frac{M(y)}{1 - y} \int_{t_{k-1}}^t \frac{\partial u(x_j, s)}{\partial s} e^{-\frac{y}{\tau} \sigma(t_k - s)} ds + R \\
= \frac{M(y)}{1 - y} \sum_{i=1}^{k-1} (u^k_j - u^{i+1}_j) w_k + R \\
= \frac{M(y)}{1 - y} \left( w_k u^k_j - \sum_{i=1}^{k-1} (w_{k,i+1} - w_{k,i}) u^i - w_{k,i+1} u^i \right) + R,
\]

(24)
where
\[ w_j = e^{-\sigma t_j} - e^{-\sigma (j+1)} . \] (25)

**Theorem 1.** Suppose \( \nu(t) \in C^2[0, t_k] \). Let
\[ A = \sum_{i=1}^{k} \int_{t_{i-1}}^{t_i} \left[ \nu'(s) (s - t_{i-1}) \right] e^{-\sigma (s - t_{i-1})} ds, \]
\[ 65 \leq 71 \left( \Delta t \right) \frac{\max}\limits_{0 \leq s \leq t_i} \nu'(s) \left| 1 + \frac{\sigma^2 \Delta t}{12} + \cdots + \frac{\sigma^3 \Delta t}{24} \right| t_k. \] (27)

**Proof.** Using the Taylor series expansion with integral remainder, we have
\[ A = \frac{1}{\Delta t} \sum_{i=1}^{k} \int_{t_{i-1}}^{t_i} \left[ \nu'(s) (s - t_{i-1}) \right] e^{-\sigma (s - t_{i-1})} ds \\
- \int_{t_{i-1}}^{t_i} \nu'(s) (s - t_{i-1}) e^{-\sigma (s - t_{i-1})} ds, \]
\[ \frac{1}{\sigma \Delta t} \sum_{i=1}^{k} \int_{t_{i-1}}^{t_i} \left[ (s - t_{i-1}) e^{-\sigma (s - t_{i-1})} - \frac{\sigma^2 \Delta t}{12} \cdots - \frac{\sigma^3 \Delta t}{24} \right] t_k. \] (28)

Since,
\[ \int_{t_{i-1}}^{t_i} \left[ (s - t_{i-1}) e^{-\sigma (s - t_{i-1})} - \frac{\sigma^2 \Delta t}{12} \cdots - \frac{\sigma^3 \Delta t}{24} \right] ds, \]
\[ \leq (\Delta t)^4 \frac{1}{12} \frac{\sigma^2 \Delta t}{12} \cdots - \frac{\sigma^3 \Delta t}{24} \cdots . \] (29)

and \( e^{-\Delta t (k-j)} \leq 1 \) for \( i = 1, 2, \ldots, k \) and \( t_k = k \Delta t \); hence, the result will be achieved.

Now, using Theorem 1, we obtain
\[ M(y) \sum_{i=1}^{k} \int_{t_{i-1}}^{t_i} \left( \frac{\partial u}{\partial s} (x_i, \tau) \right) e^{-\sigma (t - t_{i-1})} ds, \]
\[ \leq \frac{M(y)}{1 - y} \frac{r^2}{\sigma \max}\limits_{0 \leq s \leq t_i} \nu'(s) \left| 1 + \frac{\sigma^2 \Delta t}{12} + \cdots + \frac{\sigma^3 \Delta t}{24} \right| t_k. \] (30)

We introduce some lemmas which will be used in numerical scheme and stability analysis.  

**Lemma 1** (see [41]). Suppose \( u(t) \in C^3[0, t_{k+1}] \); then, we have
\[ \frac{D^k_t u}{D^k_t u_{t_{k+1}}^2} - T^k_t u_{t_{k+1}} = O(t^2), \]
\[ 0 \leq k \leq N - 1. \] (31)

where
\[ T^k_t u_{t_{k+1}} = \frac{1}{\gamma t} \left( u_{j+1}^1 + u_{j+1}^2 - \sum_{i=1}^{k} \left( w_{k-i} - w_{k-i+1} \right) u_{j+1}^2 - w_k u_j \right) \]
\[ 0 \leq k \leq N - 1. \] (32)

**Lemma 2** (see [42]). For the definition \( M, \) we have \( M > 0 \) and \( M_{j+1} < M_j \forall j \leq k. \)

Now, we present the numerical scheme for solving \( 4 \)–\( 7 \) based on the Crank–Nicolson method and cubic B-spline quasi-interpolant. We approximate equations (4) and (5) at the point \( (x_j, t_{k+1}) \) by the Crank–Nicolson finite difference approximation:
\[ \frac{\partial^2 u}{\partial x^2} (x_j, t_{k+1}) - \frac{1}{2} \left[ u_{x x} (x_j, t_{k+1}) + u_{x x} (x_j, t_k) \right] \\
+ \frac{\eta}{2} \left[ u (x_j, t_{k+1}) u_{x x} (x_j, t_{k+1}) + u (x_j, t_k) u_{x x} (x_j, t_k) \right] \\
+ \frac{\alpha_1}{2} \left[ (u v)_{x x}^{k+1} + (u v)_{x x}^k \right] = q_l (x_j, t_{k+1}), \] (33)
\[ \frac{\partial^2 u}{\partial x^2} (x_j, t_{k+1}) - \frac{1}{2} \left[ u_{x x} (x_j, t_{k+1}) + u_{x x} (x_j, t_k) \right] \\
+ \frac{\eta}{2} \left[ v (x_j, t_{k+1}) v_{x x} (x_j, t_{k+1}) + v (x_j, t_k) v_{x x} (x_j, t_k) \right] \\
+ \frac{\alpha_1}{2} \left[ (u v)_{x x}^{k+1} + (u v)_{x x}^k \right] = q_l (x_j, t_{k+1}). \] (34)

The nonlinear terms in equations (33) and (34) are linearized using the following quasi-linearization [43]:
\[ u_{x x}^{k+1} = u_{x x}^{k+1} + u_{x x}^k + u_{x x}^k - (u v)^k + O(t^2), \] (35)
\[ (u v)_{x x}^{k+1} = u (x_j, t_{k+1}) v_{x x}^{k+1} + u (x_j, t_k) v_{x x}^k - (u v)^k + O(t^2), \] (36)
\[ u_{x x}^{k+1} = u (x_j, t_{k+1}) v_{x x}^{k+1} + u (x_j, t_k) v_{x x}^k - (u v)^k + O(t^2), \] (37)
\[ (u v)_{x x}^{k+1} = u (x_j, t_{k+1}) v_{x x}^{k+1} + u (x_j, t_k) v_{x x}^k - (u v)^k + O(t^2). \] (38)

Now, using (22), (24), Lemma 1, and relations (36)–(38), we have the following difference scheme which is accurate of the order \( O(t^2 + h^2) \):
\[
\left[\frac{w_0}{2} + \frac{\gamma_\eta}{2h} \sum_{l=0}^{M} D_{jl}^1 U_j^k + \frac{\alpha_1 \gamma_T}{2h} \sum_{l=0}^{M} D_{jl}^1 V_j^k\right] U_j^{k+1} - \frac{\gamma_T}{2h^2} \sum_{l=0}^{M} D_{jl}^2 U_j^k + \frac{\gamma_\eta}{2h} \sum_{l=0}^{M} D_{jl}^2 V_j^k + v_k U_j^0
\]
\[
+ \frac{\alpha_1 \gamma_T}{2h} \sum_{l=0}^{M} D_{jl}^1 V_j^k + \frac{\alpha_2 \gamma_T}{2h} \sum_{l=0}^{M} D_{jl}^1 V_j^k
\]
\[
= \gamma_T q_{1j}^{k(1/2)} - \frac{w_0}{2} U_j^k - \frac{\gamma_T}{2h} \sum_{l=0}^{M} D_{jl}^2 U_j^k + \frac{\gamma_T}{2h} \sum_{l=0}^{M} D_{jl}^2 V_j^k + v_k U_j^0
\]
\[
+ \frac{\alpha_1 \gamma_T}{2h} \sum_{l=0}^{M} D_{jl}^1 V_j^k + \frac{\alpha_2 \gamma_T}{2h} \sum_{l=0}^{M} D_{jl}^1 V_j^k
\]
\[
= \gamma_T q_{2j}^{k(1/2)} - \frac{w_0}{2} U_j^k - \frac{\gamma_T}{2h} \sum_{l=0}^{M} D_{jl}^2 U_j^k + \frac{\gamma_T}{2h} \sum_{l=0}^{M} D_{jl}^2 V_j^k + v_k V_j^0
\]
\[
+ \frac{\alpha_1 \gamma_T}{2h} \sum_{l=0}^{M} D_{jl}^1 U_j^k + \frac{\alpha_2 \gamma_T}{2h} \sum_{l=0}^{M} D_{jl}^1 V_j^k
\]

Therefore, in each time step we solve the following b ipentadiagonal linear system of dimension \((2M - 2) \times (2M - 2)\):

\[
AM^{k+1} = F_k
\]

where \(M = [U_1^{k+1}, \ldots, U_{M-1}^{k+1}, V_1^{k+1}, \ldots, V_{M-1}^{k+1}]^T\).

### 4. Stability Analysis

to study the stability analysis of the proposed scheme, we use the Fourier method. In applying the Fourier stability method, the nonlinear terms are temporarily frozen, since the stability analysis is strictly only applicable to linear equations. Thus, we have linearized the nonlinear terms \(u u_x\) and \((\nu v)_{xx}\) in equation (4) by freezing \(u\) and \(v\) as a local constants \(\beta_1\) and \(\beta_2\), respectively. We have

\[
\frac{\partial^2}{\partial x^2} U_x^{k+1} - \frac{1}{2} \left[ (U_{xx})^{k+1} + (U_{xx})^0 \right] + \frac{\eta \beta_1}{2} \left[ (U_{xx})^{k+1} + (U_{xx})^0 \right] + \frac{\alpha_1 \beta_1}{2} \left[ (V_{xx})^{k+1} + (V_{xx})^0 \right] = q_1(x, t_{k+1/2}).
\]

Substituting approximations (22) and (32) yield the following difference equation:

\[
\left[\frac{w_0}{2} U_j^{k+1} - \frac{\gamma_T}{2h^2} \sum_{l=0}^{M} D_{jl}^2 U_j^k + \frac{\gamma_T}{2h} \sum_{l=0}^{M} D_{jl}^2 V_j^k + v_k U_j^0 \right] + \frac{\alpha_2 \gamma_T}{2h} \sum_{l=0}^{M} D_{jl}^1 U_j^k
\]
\[
= \gamma_T q_{2j}^{k(1/2)} - \frac{w_0}{2} U_j^k - \frac{\gamma_T}{2h} \sum_{l=0}^{M} D_{jl}^2 U_j^k + \frac{\gamma_T}{2h} \sum_{l=0}^{M} D_{jl}^2 V_j^k + v_k V_j^0
\]
\[
+ \frac{\alpha_2 \gamma_T}{2h} \sum_{l=0}^{M} D_{jl}^1 U_j^k + \frac{\alpha_2 \gamma_T}{2h} \sum_{l=0}^{M} D_{jl}^1 V_j^k
\]

Let \(U_j^k\) and \(V_j^k\) be the approximate solutions of (43), and define \(\zeta_j^k = U_j^k - \bar{U}_j^k, \xi_j^k = V_j^k - \bar{V}_j^k, 1 \leq j \leq M - 1, 0 \leq k \leq N - 1\), with corresponding vectors:

\[
\zeta^k = (\zeta_1^k, \zeta_2^k, \ldots, \zeta_{M-1}^k)^T, \quad \xi^k = (\xi_1^k, \xi_2^k, \ldots, \xi_{M-1}^k)^T.
\]

So, we have
\[
\begin{aligned}
&\frac{w_0}{2} + \frac{\nu r}{2h^2} \left( \frac{\nu k^1}{6} - \frac{\nu k^2}{3} + \frac{\nu k^3}{3} - \frac{\nu k^4}{6} \right) + \frac{y r}{2} \left( \frac{\nu k^1}{12} - \frac{\nu k^2}{3} + \frac{\nu k^3}{3} - \frac{\nu k^4}{12} \right) \\
&+ \frac{y r a_1 b_1}{2h} \left( \frac{1}{3} + \frac{1}{2} - \frac{1}{3} \right) = \frac{w_0 k^1}{2} + \sum_{i=0}^{M} \left( w_{j-k} - w_{j-k+1} \right) \frac{t_j^i + t_{j-1}^{i-1}}{2} \\
&+ \frac{y r a_1 b_1}{2h} \left( \frac{1}{3} + \frac{1}{2} - \frac{1}{3} \right) - \frac{y r a_1 b_1}{2h} \left( \frac{1}{3} + \frac{1}{2} - \frac{1}{3} \right) + w_0 k^0,
\end{aligned}
\]  

(45)

where \( \nu = (\eta \beta_1 + \alpha_1 \beta_2) \).

Now, we define the grid functions as follows:

\[
\begin{aligned}
\zeta^k(x) &= \begin{cases}
\zeta^k_{j-1}, & x_j - \frac{h}{2} < x < x_j + \frac{h}{2} \\
0, & 0 \leq x \leq \frac{h}{2} \text{ or } L - \frac{h}{2} < x \leq L,
\end{cases} \\
\xi^k(x) &= \begin{cases}
\xi^k_{j-1}, & x_j - \frac{h}{2} < x < x_j + \frac{h}{2} \\
0, & 0 \leq x \leq \frac{h}{2} \text{ or } L - \frac{h}{2} < x \leq L.
\end{cases}
\end{aligned}
\]  

(46)

We expand the \( \zeta^k(x) \) and \( \xi^k(x) \) into the following Fourier series expansions:

\[
\begin{aligned}
\zeta^k(x) &= \sum_{l=-\infty}^{\infty} A_k(l) e^{(2\pi l x/L)}, \\
\xi^k(x) &= \sum_{l=-\infty}^{\infty} B_k(l) e^{(2\pi l x/L)},
\end{aligned}
\]  

(47)

where

\[
\begin{aligned}
A_k(l) &= \frac{1}{L} \int_0^L \zeta^k(x) e^{(-2\pi l x/L)} dx, \\
B_k(l) &= \frac{1}{L} \int_0^L \xi^k(x) e^{(-2\pi l x/L)} dx.
\end{aligned}
\]  

(48)

Applying the Parseval equality,

\[
\begin{aligned}
\int_0^L \| \zeta^k(x) \|^2 dx &= \sum_{l=-\infty}^{\infty} \| A_k(l) \|^2, \\
\int_0^L \| \xi^k(x) \|^2 dx &= \sum_{l=-\infty}^{\infty} \| B_k(l) \|^2.
\end{aligned}
\]  

(49)

we have

\[
\begin{aligned}
\| \zeta^k \|^2 &= \sum_{l=-\infty}^{\infty} \| A_k(l) \|^2, \\
\| \xi^k \|^2 &= \sum_{l=-\infty}^{\infty} \| B_k(l) \|^2.
\end{aligned}
\]  

(50)

Now, we suppose that

\[
\begin{aligned}
\zeta^k_j &= A_k e^{i\sigma_j}, \\
\xi^k_j &= B_k e^{i\sigma_j},
\end{aligned}
\]  

(52)

where \( \sigma_j = (2\pi j/L) \). Substituting the above relations into (45) leads to

\[
\begin{aligned}
&\left( \frac{w_0}{2} + \frac{\nu r}{6h^2} \cos(2\pi xh) - \frac{5\nu r}{3h} \cos(\sigma_j h) \right) A_{k+1} + i \left( \frac{\nu r a_1 b_1}{12h} \sin(2\pi xh) + \frac{5\nu r a_1 b_1}{3h} \sin(\sigma_j h) \right) B_{k+1} \\
&+ i \left( \frac{\nu}{6} \sin(2\pi xh) + \frac{4\nu}{3} \sin(\sigma_j h) \right) A_{k+1} = \left( \frac{w_0}{2} - \frac{\nu r}{6h^2} \cos(2\pi xh) + \frac{5\nu r}{3h^2} \cos(\sigma_j h) - \frac{3\nu r}{2h} \right) A_k \\
&+ w_k A_0 + i \left( \frac{\nu}{6} \sin(2\pi xh) - \frac{4\nu}{3} \sin(\sigma_j h) \right) A_k + i \left( \frac{\nu r a_1 b_1}{12h} \sin(2\pi xh) + \frac{5\nu r a_1 b_1}{3h} \sin(\sigma_j h) \right) B_k \\
&+ \sum_{i=1}^{k} \left( w_{k-i} - w_{k-i+1} \right) \frac{A_i + A_{i-1}}{2}.
\end{aligned}
\]  

(53)
Set
\[ X = \frac{w_0 + \frac{yr}{6h^2} \cos(2\sigma, h)}{2} - \frac{5yr}{3h^2} \cos(\sigma, h), \]
\[ Y = -\frac{v}{6} \sin(2\sigma, h) + \frac{4v}{3} \sin(\sigma, h), \]
\[ Z = -\frac{yr\alpha_{\beta_1}}{12h} \sin(2\sigma, h) + \frac{2yr\alpha_{\beta_1}}{3h} \sin(\sigma, h). \]
We have
\[ (X + iY)A_{k+1} + iZB_{k+1} = -(X + iY)A_k - iZB_k \]
\[ + \sum_{i=1}^{k} (w_{k-i} - w_{k-i+1}) \frac{A_i + A_{i-1}}{2} + w_k A_0 \]
so that
\[ |A_{k+1}| \leq |A_k| + \frac{|iZ|}{|X + iY|} \left(|B_{k+1}| + |B_k|\right) \]
\[ + \sum_{i=1}^{k} (w_{k-i} - w_{k-i+1}) \frac{A_i + A_{i-1}}{2} \left(\frac{1}{|X + iY|}\right) + w_k |A_0| \cdot \frac{|X + iY|}{|X + iY|} \]
\[ |A_{k+1}| \leq |A_k| + \frac{|iZ|}{|X + iY|} \left(|B_{k+1}| + |B_k|\right) \]
\[ + \sum_{i=1}^{k} (w_{k-i} - w_{k-i+1}) \frac{A_i + A_{i-1}}{2} \left(\frac{1}{|X + iY|}\right) + w_k |A_0| \cdot \frac{|X + iY|}{|X + iY|} \]
\[ \text{(55)} \]

Theorem 2. If \( A_k \) is the solution of equation (55), then there are positive constants \( C_k \) such that
\[ |A_k| \leq C_k |A_0|, \quad k = 1, 2, \ldots, N - 1. \]

Proof. We use the mathematical induction for proof. For \( k = 1 \), we have
\[ |A_1| \leq |A_0| + \frac{|iZ|}{|X + iY|} \left(|B_1| + |B_0|\right) + w_0 |A_0| \cdot \frac{|X + iY|}{|X + iY|} \]
\[ \text{(58)} \]
Using the convergence of the series on the right-hand side of equation (49), we know that there exists a positive constant \( P_2 \) such that
\[ |B_k| \leq P_2 |A_0|, \quad k = 0, 1, \ldots, N - 1. \]
So,
\[ |A_k| \leq P_2 |A_0| + D(|A_0|) + E|A_0| \leq C_1 |A_0|. \]
\[ \text{(60)} \]
Now, suppose that
\[ |A_k| \leq C_k |A_0|, \quad k = 1, 2, \ldots, N - 2. \]
By equation (56), we have
\[ |A_{k+1}| \leq C_k |A_0| + FC_2 |A_0| + G \sum_{i=1}^{k} (w_{k-i} - w_{k-i+1}) \]
\[ + \frac{C_1 |A_0| + C_2 |A_0|}{2} + H |A_0|. \]
\[ \text{(62)} \]
Now, assume that
\[ C' = \max\{C_1, C_2, \ldots, C_{N-2}\}. \]
\[ \text{(63)} \]
So,
\[ |A_{k+1}| \leq C_k |A_0| + FP_2 |A_0| + GC' |A_0| + H |A_0| \leq C_{k+1} |A_0|. \]
\[ \text{(64)} \]
Remark 1. From (5), with similar way, there are positive constants \( Q_k \) such that
\[ |B_k| \leq Q_k |B_0|, \quad k = 1, 2, \ldots, N - 1. \]
\[ \text{(65)} \]

Theorem 3. The finite difference schemes (39) and (40) are unconditionally stable for \( \gamma \in (0, 1) \).

Proof. According to Theorem 2 and Remark 1, using (51), we obtain
\[ \| \xi^k \|_2 \leq \sum_{l=-\infty}^{\infty} \| A_k (l) \|_2 \leq \sum_{l=-\infty}^{\infty} C_k \| A_0 (l) \|_2 = C_k \| \xi^0 \|_2, \]
\[ \| \eta^k \|_2 \leq \sum_{l=-\infty}^{\infty} \| B_k (l) \|_2 \leq \sum_{l=-\infty}^{\infty} Q_k \| B_0 (l) \|_2 = Q_k \| \eta^0 \|_2, \]
so that
\[ \| U^k - \tilde{U}^0 \|_2 \leq C_k \| \tilde{U}^0 - \tilde{U}^0 \|_2, \]
\[ \| V^k - \tilde{V}^0 \|_2 \leq Q_k \| \tilde{V}^0 - \tilde{V}^0 \|_2, \]
which shows that schemes (39) and (40) are unconditionally stable. \( \square \)

5. Numerical Results
In this section, we provide two examples to illustrate efficiency of schemes (39) and (40). All experiments are performed on a Windows 10 (64-bit) Intel(R) Core(TM) i7-7500U CPU 2.70 GHz, 8.0 GB of RAM using MATLAB R2017b. In all examples, we use the error norm
\[ \| e(\tau, h) \| = \| e^{\gamma} \| = \left( \frac{h \sum_{j=1}^{M} (e^{\gamma})^2}{(1/2)} \right), \]
\[ \text{(68)} \]
where \( e^\gamma = u^\gamma - U^\gamma \) and \( e^\gamma = V^\gamma - \tilde{V}^\gamma \). We evaluate the convergence order with the following formula:
\[ r(\tau, h) = \log_2 \left( \frac{\| e(\tau, 2h) \|}{\| e(\tau, h) \|} \right). \]
\[ \text{(69)} \]

Example 1. Consider the coupled Burgers equation with time fractional derivative with exact solutions \( u(x, t) = v(x, t) = e^{x^2} (1 - x)^2 \):
We derive the functions $q_1$ and $q_2$ with the help from exact solutions. Numerical solution and exact solution have been demonstrated in Figure 1. Tables 1–4 give the approximation errors and CPU times for the different schemes. We choose different space step sizes to obtain the numerical results and order of convergence. Figure 2 shows the comparison of numerical solution and exact solution for $\gamma = 0.9$ at $t = 1$ and contour plot of numerical solution. Also, Figure 3 shows the pointwise errors for $u(x,t)$.

**Example 2.** Consider the following coupled Burgers equation with time fractional derivative with exact solutions $u(x,t) = v(x,t) = x(1-x)\sin(t)$:

\[
\begin{cases}
\frac{\partial^\alpha u}{\partial t^\alpha} - \frac{\partial^2 u}{\partial x^2} + \eta \frac{\partial u}{\partial x} + a_1 \frac{\partial (uv)}{\partial x} = q_1, & x \in [0,1], t \in [0,T], 0 < \gamma < 1, \\
\frac{\partial^\alpha v}{\partial t^\alpha} - \frac{\partial^2 v}{\partial x^2} + \eta \frac{\partial v}{\partial x} + a_2 \frac{\partial (uv)}{\partial x} = q_2, & x \in [0,1], t \in [0,T], 0 < \gamma < 1, \\
u(0,t) = 0, v(0,t) = 0, \\
u(1,t) = 0, v(1,t) = 0, \\
u(x,0) = x^2(1-x)^2, v(x,0) = x^2(1-x)^2. 
\end{cases}
\]

(70)

We solve this problem with the method developed in this article with several values of $T$ and $\gamma$. Graphs of numerical solution and exact solution at different times have been demonstrated in Figure 4. Figure 4 shows that the proposed method is efficient. Table 5 gives the approximation errors for $t = 0.5, 1, 1.5, 2, 2.5, 3, 4$ with different $a_1, a_2$ and CPU times. We choose $h = (1/160)$ for space step size to obtain the numerical results. Figure 5 shows the comparison of numerical solution and exact solution for $\gamma = 0.1$ at $t = 1$. Also, Figure 6 shows the pointwise errors and contour plot of numerical solution for $u(x,t)$.
### Table 1: \( \tau = 0.001 \) and \( \eta = 800 \) and \( \alpha_1 = 0.01, \alpha_2 = 0.01 \) experiment order of convergence for \( u(x, t) \) at \( \gamma = 0.1, 0.3, 0.5 \) for Example 1.

| \( h \)   | \( \|e^N\| \) | Order | CPU         | \( \|e^N\| \) | Order | CPU             | \( \|e^N\| \) | Order | CPU             |
|-----------|----------------|-------|-------------|----------------|-------|-----------------|----------------|-------|-----------------|
| (1/10)    | 1.116e−03      | 1.6557s | 1.1133e−03 | 1.4701s        | 1.1094e−03 | 1.5101s         |
| (1/20)    | 3.8484e−04     | 2.5941s | 2.8074s     | 2.5966s        | 2.7374s   | 2.6013s         |
| (1/40)    | 4.7604e−05     | 1.9571s | 5.4495s     | 1.9574s        | 5.2651s   | 1.9579s         |
| (1/80)    | 1.1883e−05     | 2.0022s | 10.6861s    | 2.0018s        | 10.6804s  | 2.0011s         |
| (1/160)   | 2.7784e−06     | 2.0966s | 22.2362s    | 2.0018s        | 23.9606s  | 2.0944s         |

### Table 2: \( \tau = 0.001 \) and \( \eta = 800 \) and \( \alpha_1 = 0.01, \alpha_2 = 0.01 \) experiment order of convergence for \( u(x, t) \) at \( \gamma = 0.7, 0.9 \) for Example 1.

| \( h \)   | \( \|e^N\| \) | Order | CPU         | \( \|e^N\| \) | Order | CPU             | \( \|e^N\| \) | Order | CPU             |
|-----------|----------------|-------|-------------|----------------|-------|-----------------|----------------|-------|-----------------|
| (1/10)    | 1.1018e−03     | 1.6541s | 1.0653e−03 | 1.7714s        |
| (1/20)    | 1.8045e−04     | 2.6102s | 1.7495e−04 | 2.6062s        |
| (1/40)    | 4.6426e−05     | 1.9586s | 4.5085e−05 | 4.7604s        |
| (1/80)    | 1.1614e−05     | 1.9991s | 1.1364e−05 | 1.1547s        |
| (1/160)   | 2.7286e−06     | 2.0896s | 2.7292e−06 | 3.15002s       |

### Table 3: \( \tau = 0.001 \) and \( \eta = 200 \) and \( \alpha_1 = 0.03, \alpha_2 = 0.03 \) experiment order of convergence for \( v(x, t) \) at \( \gamma = 0.1, 0.3, 0.5 \) for Example 1.

| \( h \)   | \( \|e^N\| \) | Order | CPU         | \( \|e^N\| \) | Order | CPU             | \( \|e^N\| \) | Order | CPU             |
|-----------|----------------|-------|-------------|----------------|-------|-----------------|----------------|-------|-----------------|
| (1/10)    | 1.1129e−03     | 1.6208s | 1.1110e−03 | 1.5264s        |
| (1/20)    | 1.8595e−04     | 2.5813s | 1.8514e−04 | 2.5889s        |
| (1/40)    | 4.7628e−05     | 1.9650s | 4.7411e−05 | 4.5895s        |
| (1/80)    | 1.1827e−05     | 2.0097s | 1.1775e−05 | 2.0088s        |
| (1/160)   | 2.7407e−06     | 2.1095s | 2.7296e−06 | 2.1079s        |

### Table 4: \( \tau = 0.001 \) and \( \eta = 200 \) and \( \alpha_1 = 0.03, \alpha_2 = 0.03 \), experiment order of convergence for \( v(x, t) \) at \( \gamma = 0.7, 0.9 \) for Example 1.

| \( h \)   | \( \|e^N\| \) | Order | CPU         | \( \|e^N\| \) | Order | CPU             |
|-----------|----------------|-------|-------------|----------------|-------|-----------------|
| (1/10)    | 1.0983e−03     | 1.6151s | 1.0614e−03 | 1.6646s        |
| (1/20)    | 1.8136e−04     | 2.5983s | 3.0172s     | 3.3234s        |
| (1/40)    | 4.6406e−05     | 1.9665s | 6.1189s     | 6.7189s        |
| (1/80)    | 1.1544e−05     | 2.0072s | 13.5179s    | 13.1345s       |
| (1/160)   | 2.6855e−06     | 2.1039s | 25.4015s    | 31.4012s       |
Figure 2: The comparison (a) and absolute error (b) between numerical solution and exact solution with $\tau = (1/1000)$, $h = (1/160)$, and $\gamma = 0.9$ at $t = 1$ for Example 1.

Figure 3: Pointwise errors for $u(x, t)$ with $\tau = (1/1000)$, $h = (1/160)$, and $\gamma = 0.9$ (a) and contour plot of numerical solution $u(x, t)$ (b) for Example 1.

Figure 4: Graphs of numerical and exact solution at different times with $\tau = 0.001$ for Example 2.
6. Conclusion

In this article, we constructed a Crank–Nicolson finite difference scheme based on cubic B-spline quasi-interpolation to solve time fractional coupled Burgers equations. By the Fourier series method, we proved that this scheme is unconditionally stable. Numerical examples have been carried out to show the convergence orders and applicability of the scheme and error norms are calculated with respect to different space step sizes. From the error tables and graphs of exact and numerical solution, we can say that our method has a good accuracy. For the numerical computations, we have used Matlab.

Data Availability

All results have been obtained by conducting the numerical procedure and the ideas can be shared for the researchers.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Table 5: Errors for \( u(x,t) \) and \( v(x,t) \) at different times for Example 2.

| \( T \) | \( \alpha_1 \) | \( \alpha_2 \) | \( \gamma = 0.1 \) | \( \gamma = 0.3 \) | \( \gamma = 0.7 \) |
|-------|--------|--------|----------------|----------------|----------------|
|       |        |        | \( \|e_N^u\| \) | \( \|e_N^v\| \) | CPU           | \( \|e_N^u\| \) | \( \|e_N^v\| \) | CPU           | \( \|e_N^u\| \) | \( \|e_N^v\| \) | CPU           |
| 0.5   | 0.01   | 0.01   | 2.2750e-07    | 1.0966e-08    | 19.7143s       | 1.0692e-07    | 19.6825s      | 2.1196e-07    | 2.3957e-09    | 22.1880s      |
| 1     | 0.03   | 0.02   | 2.1872e-06    | 1.9266e-08    | 19.7329s       | 1.8640e-08    | 19.9076s      | 2.0966e-06    | 5.8427e-09    | 24.0896s      |
| 1.5   | 0.001  | 0.005  | 8.1009e-08    | 2.2678e-08    | 19.6884s       | 2.1703e-08    | 20.8275s      | 9.8131e-08    | 6.6570e-09    | 25.0396s      |
| 2     | 0.001  | 0.001  | 6.6249e-08    | 2.0562e-08    | 19.8765s       | 1.9294e-08    | 21.4610s      | 9.1599e-08    | 3.4789e-09    | 25.4790s      |
| 2.5   | 0.002  | 0.002  | 6.3910e-08    | 1.3390e-08    | 19.9632s       | 1.9531e-08    | 22.1602s      | 7.207e-07     | 4.5790e-08    | 25.9242s      |
| 3     | 0.01   | 0.01   | 3.3597e-08    | 2.8784e-09    | 19.9462s       | 1.4200e-09    | 22.4854s      | 1.2871e-09    | 28.0649s      |
| 4     | 0.01   | 0.01   | 6.2662e-07    | 1.7635e-08    | 20.2347s       | 1.8725e-09    | 23.4227s      | 6.0534e-07    | 28.6413s      |

Figure 5: The plot of exact solution (a) and numerical solution (b) at \( \tau = 1/1000 \) and \( h = (1/160) \) with \( \gamma = 0.1 \) for Example 2.

Figure 6: Pointwise errors for \( v(x,t) \) with \( \tau = (1/1000), h = (1/160), \) and \( \gamma = 0.1 \) (a) and contour plot of numerical solution \( u(x,t) \) (b) for Example 2.
References

[1] S. E. Esipov, "Coupled Burgers equations: a model of poly-dispersive sedimentation," *Physical Review E*, vol. 52, no. 4, p. 3711, 1995.

[2] J. D. Logan, *An Introduction to Nonlinear Partial Differential Equations*, Vol. 89, John Wiley & Sons, Hoboken, NJ, USA, 2008.

[3] R. C. Mittal and G. Arora, "Numerical solution of the coupled viscous Burgers’ equation," *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, no. 3, pp. 1304–1313, 2011.

[4] I. J. Schoenberg, "Contributions to the problem of approximation of equidistant data by analytic functions. part B. on the problem of osculatory interpolation, a second class of analytic approximation formulas," *Quarterly of Applied Mathematics*, vol. 4, no. 2, pp. 112–141, 1946.

[5] M. J. Siddiqui, I. Muhi, M. Abbas, and T. Akram, "Nonlinear waves propagation and stability analysis for planar waves at far field using quintic B-spline collocation method," *Alexandria Engineering Journal*, vol. 59, no. 4, pp. 2695–2703, 2020.

[6] M. Amin, M. Abbas, M. K. Iqbal, and D. Baleanu, "Numerical treatment of time-fractional Klein–Gordon equation using redefined extended cubic B-spline functions," *Frontiers in Physics*, vol. 8, 2020.

[7] M. K. Iqbal, M. Abbas, and I. Wasim, "New cubic B-spline approximation for solving third order Emden-Flower type equations," *Applied Mathematics and Computation*, vol. 331, pp. 319–333, 2018.

[8] N. Khalid, M. Abbas, and M. K. Iqbal, "Non-polynomial quintic spline for solving fourth-order fractional boundary value problems involving product terms," *Applied Mathematics and Computation*, vol. 349, pp. 393–407, 2019.

[9] T. Nazir, M. Abbas, and M. K. Iqbal, "New cubic B-spline approximation technique for numerical solutions of Coupled Viscous Burgers equations," *Engineering Computations*, vol. 2020, In press.

[10] M. Amin, M. Abbas, M. K. Iqbal, A. I. M. Ismail, and D. Baleanu, "A fourth order non-polynomial quintic spline collocation technique for solving time fractional super-diffusion equations," *Advances in Difference Equations*, vol. 2019, no. 1, p. 514, 2019.

[11] N. Khalid, M. Abbas, M. K. Iqbal, and D. Baleanu, "A numerical investigation of Caputo time fractional Allen–Cahn equation using redefined cubic B-spline functions," *Advances in Difference Equations*, vol. 2020, no. 1, pp. 1–22, 2020.

[12] T. Akram, M. Abbas, A. Iqbal, D. Baleanu, and J. H. Asad, "Novel numerical approach based on modified extended cubic B-spline functions for solving non-linear time-fractional telegraph equation," *Symmetry*, vol. 12, no. 7, p. 1154, 2020.

[13] T. Akram, M. Abbas, A. Ali, A. Iqbal, and D. Baleanu, "A numerical approach of a time fractional reaction-diffusion model with a non-singular kernel," *Symmetry*, vol. 12, no. 10, p. 1653, 2020.

[14] T. Akram, M. Abbas, M. B. Riaz, A. I. Ismail, and N. M. Ali, "An efficient numerical technique for solving time fractional Burgers equation," *Alexandria Engineering Journal*, vol. 59, no. 4, 2020.

[15] N. Khalid, M. Abbas, M. K. Iqbal, J. Singh, and A. I. M. Ismail, "A computational approach for solving time fractional differential equation via spline functions," *Alexandria Engineering Journal*, vol. 59, no. 5, pp. 3061–3078, 2020.

[16] M. Caputo and M. Fabrizio, "A new definition of fractional derivative without singular kernel," *Progress in Fractional Differentiation and Applications*, vol. 1, no. 2, pp. 1–13, 2015.

[17] S. Noieaghdam and D. Sidorov, "Caputo-Fabrizio fractional derivative to solve the fractional model of energy supply-demand system," *International Information and Engineering Technology Association*, vol. 7, no. 3, pp. 359–367, 2020.

[18] H. Günerhan, H. Dutta, M. A. Dokuyucu, and W. Adel, "Analysis of a fractional HIV model with Caputo and constant proportional Caputo operators," *Chaos, Solitons & Fractals*, vol. 139, Article ID 110053, 2020.

[19] D. Baleanu, A. Jajarmi, H. Mohammadi, and S. Rezapour, "A new study on the mathematical modelling of human liver with Caputo-Fabrizio fractional derivative," *Chaos, Solitons & Fractals*, vol. 134, Article ID 109705, 2020.

[20] K. A. Abro and A. Atangana, "A comparative analysis of electromechanical model of piezoelectric actuator through Caputo–Fabrizio and Atangana–Baleanu fractional derivatives," *Mathematical Methods in the Applied Sciences*, vol. 43, no. 3, 2020.

[21] X. Zheng, H. Wang, and H. Fu, "Well-posedness of fractional differential equations with variable-order Caputo-Fabrizio derivative," *Chaos, Solitons & Fractals*, vol. 138, Article ID 109666, 2020.

[22] X. Gong and M. A. Khan, "A new numerical solution of the competition model among bank data in Caputo-Fabrizio derivative," *Alexandria Engineering Journal*, vol. 59, no. 4, pp. 2251–2259, 2020.

[23] S. Kumar, J. F. Gómez Aguilar, and P. Pandey, "Numerical solutions for the reaction-diffusion, diffusion-wave, and Cattaneo equations using a new operational matrix for the Caputo-Fabrizio derivative," *Mathematical Methods in the Applied Sciences*, vol. 43, no. 15, pp. 8595–8607, 2020.

[24] A. Shaikh, A. Tassaddiq, K. S. Nisar, and D. Baleanu, "Analysis of differential equations involving Caputo–Fabrizio fractional operator and its applications to reaction–diffusion equations," *Advances in Difference Equations*, vol. 2019, no. 1. p. 178, 2019.

[25] M. Xu and Y. Jian, "Unsteady rotating electroosmotic flow with time-fractional Caputo-Fabrizio derivative," *Applied Mathematics Letters*, vol. 100, Article ID 106015, 2020.

[26] K. Liu, M. Fečkan, and J. Wang, "A fixed-point Approach to the hyers-ulam stability of caputo-fabrizio fractional differential equations," *Mathematics*, vol. 8, no. 4, p. 647, 2020.

[27] Y. Chen and H.-L. An, "Numerical solutions of coupled Burgers equations with time- and space-fractional derivatives," *Applied Mathematics and Computation*, vol. 200, no. 1, pp. 87–95, 2008.

[28] N. A. Khan, A. Ara, and A. Mahmood, "Numerical solutions of time-fractional Burgers equations," *International Journal of Numerical Methods for Heat & Fluid Flow*, vol. 22, no. 2, pp. 175–193, 2012.

[29] A. Prakash, M. Kumar, and K. K. Sharma, "Numerical method for solving fractional coupled Burgers equations," *Applied Mathematics and Computation*, vol. 260, pp. 314–320, 2015.

[30] J. Singh, D. Kumar, and R. Srivastava, "Numerical solution of time- and space-fractional coupled Burgers’ equations via homotopy algorithm," *Alexandria Engineering Journal*, vol. 55, no. 2, pp. 1753–1763, 2016.

[31] H. Aminikhah and N. Malekzadeh, "An efficient method for systems of variable coefficient Coupled Burgers’ equation with time-fractional derivative," *The Scientific World Journal*, vol. 2013, Article ID 687695, 2013.
[32] H. F. Ahmed, M. S. M. Baghat, and M. Zaki, “Analytical approaches to space-and time-fractional coupled Burgers’ equations,” Pramana, vol. 92, no. 3, p. 38, 2019.

[33] B. Albouhimad and H. Adibi, “On a hybrid spectral exponential Chebyshev method for time-fractional coupled Burgers equations on a semi-infinite domain,” Advances in Difference Equations, vol. 2017, no. 1, p. 85, 2017.

[34] T. A. Sulaiman, M. Yavuz, H. Bulut, and H. M. Baskonus, “Investigation of the fractional coupled viscous Burgers’ equation involving Mittag-Leffler kernel,” Physica A: Statistical Mechanics and its Applications, vol. 527, Article ID 121126, 2019.

[35] J. Liu and G. Hou, “Numerical solutions of the space- and time-fractional coupled Burgers equations by generalized differential transform method,” Applied Mathematics and Computation, vol. 217, no. 16, pp. 7001–7008, 2011.

[36] N. Özdemir, A. Secer, and M. Bayram, “The gegenbauer wavelets-based computational methods for the coupled system of Burgers’ equations with time-fractional derivative,” Multidisciplinary Digital Publishing Institute, vol. 7, no. 6, p. 486, 2019.

[37] L. Schumaker, Spline Functions: Basic Theory, Siam Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2015.

[38] L. Schumaker, Spline Functions: Basic Theory, Wiley-Interscience, New York, NY, USA, 1981.

[39] K. Hollig and J. Horner, Approximation and Modeling with B-Splines, SIAM, Philadelphia, PA, USA, 2013.

[40] M. Taghipour and H. Aminikhah, A new compact alternating direction implicit method for solving two dimensional time fractional diffusion equation with Caputo-Fabrizio derivative, 2020.

[41] H. Li, S. Lü, and T. Xu, “A fully discrete spectral method for fractional Cattaneo equation based on Caputo-Fabrizo derivative,” Numerical Methods for Partial Differential Equations, vol. 35, no. 3, pp. 936–954, 2019.

[42] Z. Liu, A. Cheng, and X. Li, “A second order Crank-Nicolson scheme for fractional Cattaneo equation based on new fractional derivative,” Applied Mathematics and Computation, vol. 311, pp. 361–374, 2017.

[43] S. G. Rubin and R. A. Graves, Cubic Spline Approximation for Problems in Fluid Mechanics, NASA TR R-436, Washington, DC, USA, 1975.