Hindawi
Journal of Mathematics
Volume 2023, Article ID 5493451, 9 pages
https://doi.org/10.1155/2023/5493451

Research Article

A Meshfree Time-Splitting Approach for the Time-Fractional Burgers' Equation

Erdal Korkmaz and Kenan Yildirim

Mathematics, Mus Alparslan University, Mus, Turkey

Correspondence should be addressed to Kenan Yildirim; k.yildirim@alparslan.edu.tr

Received 18 October 2022; Revised 7 December 2022; Accepted 7 January 2023; Published 21 January 2023

Academic Editor: Guotao Wang

We consider to represent an algorithm for time-fractional Burgers' equation utilizing the multiquadric-radial basis functions with the time-splitting technique. This algorithm is performed on the three examples. The numerical results indicated that the algorithm yields accurate approximate solutions of the time-fractional Burgers' equation. The physical behavior of the obtained solutions is given with graphics, and the results showed that the obtained solutions are in good match with the solutions reported in the literature. The algorithm is accurate, flexible, and easy to implement.

1. Introduction

We address the numerical solution of the one-dimensional fractional Burgers' equation as follows:

$$\frac{\partial^\alpha u}{\partial t^\alpha} + \frac{\partial}{\partial x}\left(\frac{u^2}{2}\right) = \nu\frac{\partial^2 u}{\partial x^2} + g(x,t), \quad (x,t) \in (0,1)\times(0,T], \tag{1}$$

with the initial and boundary conditions

$$\begin{aligned} u(x,0) &= u_0(x), \quad 0 \le x \le 1,\\ u(0,t) &= u_1(t),\\ u(1,t) &= u_2(t), \quad 0 \le t \le T, \end{aligned} \tag{2}$$

respectively, where $0 < \alpha \le 1$ is Caputo fractional derivative; t and x are time and space parameters, respectively; $\nu > 0$ is the viscosity constant; and $u_0(x)$, $u_1(t)$, and $u_2(t)$ are known functions, and $u_0(x)$ is sufficiently smooth. Here, $(\partial^\alpha u/\partial t^\alpha)$ are fractional derivatives in Caputo sense, defined by

$$\frac{\partial^\alpha u(x,t)}{\partial t^\alpha} \coloneqq \frac{1}{\Gamma(1-\alpha)}\int_0^t \frac{u_t(x,\tau)}{(t-\tau)^\alpha}\mathrm{d}\tau. \tag{3}$$

The steady-state solutions of the classical Burgers' equation are primarily computed in [1]. Burger [2] proposed to utilize classical Burgers' equation for the turbulence model. Since then, Burgers' equation has always been the focus of attention for researchers. Particularly, the development of the fractional derivative in recent years has required the fractional modeling of classical mathematical models. Burgers' (1) is important one of such models. The reason why fractional derivative has been a hot topic lately is that it finds application in science and engineering in fields such as physics, biology, and control theory [3, 4]. When it was difficult for researchers to find analytical solutions for fractional differential equation models, error analyzes have been made by searching numerical solutions in general. The authors of [5] presented an approximate scheme for fractional differential equations which is based on the generalization of the Adams–Bashforth–Moulton method and then provided an exhaustive error analysis of this scheme. The authors of [6] tailored a fully discrete difference scheme for a diffusion-wave system and it is based on the transforming the governing equation into a low-order system of equations by defining two new variables. In [7], the author proves that the smooth solutions exist for the Caputo-type

fractional differential equations. The author in [8] presented a scheme for the fractional differential equation. This scheme is tailored by combining the predictor-corrector scheme with the short memory principle. The authors of [9] analyzed the stability of the explicit multistep methods and derived novel schemes which enable a larger stability region. The local truncation error and the stability of the high-order compact finite difference scheme for the fractional diffusion equation are investigated in [10]. The fractional linear multistep and the finite element techniques are performed for time and space discretization, respectively, for the time-fractional subdiffusion equation in [11]. The convergence and stability analysis of the finite difference schemes for fractional differential equations are investigated in [12]. A numerical scheme based on the combination of the Alikhanov scheme and the Galerkin finite element methods for solving nonlinear time-fractional parabolic equations is proposed in [13]. A linear finite difference scheme is tailored for the numerical solutions of the generalized time-fractional Burgers equation in [14]. In [15, 16], the authors used lower- and higher-order classical time-splitting methods to solve Burgers' equation. The authors of [17] presented three different time-splitting algorithms with the convergence and stability analysis for nonlinear time-fractional differential equations. In [12–16, 18–27], the authors proposed numerical schemes such as finite difference, finite elements, spectral methods, and radial basis functions to analyze the numerical solutions of the fractional Burgers'-type equations.

In this work, we propose an algorithm depending on the utilization of the multiquadric-radial basis functions and a splitting method for solving time-fractional Burgers' equation numerically. By using advantage of a meshfree scheme in this algorithm, the goal is to get an approach that yields accurate results and exhibits the physical dynamics of the time-fractional Burgers' equation.

The remainder of this article is organized as follows. In Section 2, the multiquadric-radial basis functions (MQ-RBF) used in this paper were introduced. Section 3 contains the time-splitting technique. In Section 4, the numerical results are presented to show the performance of the proposed algorithm. Finally, we give some concluding remarks in Section 5.

2. The Multiquadric-Radial Basis Functions (MQ-RBFs)

In order to obtain a fractional-order differential equation system, we discretize equation (1) in space by performing the radial basis functions method [28, 29]. To do this, one can represent the numerical solution of equation (1), $u(x,t)$, as follows:

$$u(x,t) \approx \sum_{j=0}^{k} \beta_j(t)\theta_j(x), \quad x \in \mathbb{R}^l, \tag{4}$$

where $\theta_j(x)$ is continuously differentiable radial basis functions, k represents the number of distributed nodes, $x = (x_0, x_1, \ldots, x_{l-1})$ is l dimensional vector, and β_j are needed to be defined by utilizing the collocation methods. On the other hand, we use the following MQ-RBF function:

$$\theta_j(x) = \sqrt{\xi_j^2 + c^2}, \tag{5}$$

where c is the shape constant and $\xi_j = \|x - x_j\|$ denoted by the Euclidean norm.

We define an approximate solution $u(x,t)$ by $u^k(x,t)$ and then write

$$u^k(x,t) = \sum_{j=0}^{k} \beta_j(t)\theta_j(x) = \boldsymbol{\Theta}^T(x)\boldsymbol{\beta}, \tag{6}$$

where $\theta_j(x) = \sqrt{(x - x_j)^2 + c^2}$, $x_j = j/k$ $(j = 0, 1, \ldots, k)$ are uniformly distributed nodes on the range [0, 1], and

$$\begin{aligned} \boldsymbol{\Theta}(x) &= \left(\theta_0(x), \theta_1(x), \ldots, \theta_k(x)\right)^T, \\ \boldsymbol{\beta} &= \left(\beta_0(t), \beta_1(t), \ldots, \beta_k(t)\right)^T. \end{aligned} \tag{7}$$

Equation (6) can be rewritten in the following form:

$$\boldsymbol{\Lambda}\boldsymbol{\beta} = w, \tag{8}$$

where $w = (u_0(t), u_1(t), \ldots, u_k(t))$ with $u^k(x_i,t) = u_i(t)$ for $i = 1, \ldots, k$, and

$$\boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{\Theta}^T(x_0) \\ \boldsymbol{\Theta}^T(x_1) \\ \vdots \\ \boldsymbol{\Theta}^T(x_k) \end{bmatrix} = \begin{bmatrix} \theta_0(x_0) & \theta_1(x_0) & \ldots & \theta_k(x_0) \\ \theta_0(x_1) & \theta_1(x_1) & \ldots & \theta_k(x_1) \\ \vdots & \vdots & \ddots & \vdots \\ \theta_0(x_k) & \theta_1(x_k) & \ldots & \theta_k(x_k) \end{bmatrix}. \tag{9}$$

By considering equations (4), (6), and (8), it follows

$$\begin{aligned} u(x,t) \approx u^k(x,t) &= \sum_{j=0}^{k} \beta_j \theta_j(x) \\ &= \boldsymbol{\Theta}^T(x)\boldsymbol{\beta} \\ &= \boldsymbol{\Theta}^T(x)\boldsymbol{\Lambda}^{-1} w = \Phi(x) w. \end{aligned} \tag{10}$$

If one replaces equation (10) and its derivatives with the corresponding terms in equation (1) at the collocation nodes $x_i = (i/k)$ $(i = 0, 1, \ldots, k)$ in [0, 1], then the following is obtained:

$$\left({}^C D_0^{\alpha} w\right)(t) + w(t) * \left(\boldsymbol{\Phi}_x w(t)\right) = \nu \boldsymbol{\Phi}_{xx} 2A w(t) + g(t), \tag{11}$$

where $({}^C D_0^{\alpha} w)(t)$, $\alpha \in (0, 1)$ is the Caputo derivative, $w = (u_0, u_1, \ldots, u_k)$,

$$\begin{aligned} g(t) &= \left(g(x_0,t), g(x_1,t), \ldots, g(x_k,t)\right), \\ \boldsymbol{\Phi}_x &= \left[\frac{\partial}{\partial x}\Phi_j(x_i)\right]_{(k+1)\times(k+1)}, \\ \boldsymbol{\Phi}_{xx} &= \left[\frac{\partial^2}{\partial x^2}\Phi_j(x_i)\right]_{(k+1)\times(k+1)}. \end{aligned} \tag{12}$$

Notice that, the operator "$*$" is defined for point-wise multiplication. Furthermore, equations (11) read in the form as follows:

$$\left({}^{C}D_0^{\alpha}w\right)(t) = Aw(t) + f(t, w(t)), \tag{13}$$

where $A = \nu\Phi_{xx}$, $\quad f(t, w(t)) = g(t) - w(t) * (\Phi_x w(t))$.

3. Time-Splitting Method

Splitting methods have received much attention due to their advantages such as speed, accuracy, and stability. These methods are explicit and easy to implement. Furthermore, splitting methods are designed as geometric numerical schemes. These schemes preserve some qualitative properties of the systems [30]. Let us consider following fractional differential equation:

$$\begin{aligned}\left({}^{C}D_0^{\alpha}w\right)(t) &= \beta w(t) + \rho w(t),\\ w(0) &= w_0, \quad 0 < t \le T,\ \beta, \rho \in \mathbb{C},\end{aligned} \tag{14}$$

where $({}^{C}D_0^{\alpha}w)(t)$, $\quad \alpha \in (0,1)$ is the Caputo derivative. The classical time-splitting method for one time step $0 \le t \le h$ can be considered for equation (14) as follows:

$$\begin{aligned}\left({}^{C}D_0^{\alpha}\overline{\overline{w}}\right)(t) &= \beta\overline{\overline{w}}(t),\\ \overline{\overline{w}}(0) &= w_0,\\ \left({}^{C}D_0^{\alpha}w\right)(t) &= \rho\widetilde{w}(t),\\ \widetilde{w}(0) &= \overline{\overline{w}}(h).\end{aligned} \tag{15}$$

That is, equation (14) is split into two equations which are solvable and easy to deal with separately. These two equations are solved successively. Let us denote $\widetilde{w}$ as approximate solution of fractional equation (14) and then the error of the classical splitting methods is given as follows:

$$\begin{aligned}w(h) - \widetilde{w}(h) = w_0 &\left[\sum_{k=0}^{\infty}\frac{(\beta+\rho)^k h^{k\alpha}}{\Gamma(1+k\alpha)} - \sum_{k=0}^{\infty}\frac{(\beta h^{\alpha})^k}{\Gamma(1+k\alpha)}\right.\\ &\left.\cdot\sum_{k=0}^{\infty}\frac{(\rho h^{\alpha})^k}{\Gamma(1+k\alpha)}\right]\\ &= \left[\frac{2\beta\rho}{\Gamma(1+2\alpha)} - \frac{\beta\rho}{(\Gamma(1+\alpha))^2}\right]h^{2\alpha} + O\left(h^{3\alpha}\right).\end{aligned} \tag{16}$$

The order of convergence of the classical splitting method is $O(h^{2\alpha})$. Thus, it can be said that the convergence order of the time-splitting scheme obtained for one step decreases as α approaches zero. In order to solve this problem, the author of [17] developed a division method for the fractional nonlinear systems of ordinary differential equations. The order of the convergence of this new division method is $O(h^{1+\alpha})$ and we perform this method to solve equation (13), with $\alpha \in (0,1)$, A is $(k+1)\times(k+1)$ real matrices, $f\colon [0,T]\times R^{k+1} \longrightarrow R^{k+1}$. Here, the continuous function f fulfills the Lipschitz inequality, that is, on suitable domain G,

$$\left|f(t, w_1) - f(t, w_2)\right| \le K\left|w_1 - w_2\right|, \quad \forall w_1, w_2 \in G. \tag{17}$$

The Caputo derivative is defined by

$$\begin{aligned}\left({}^{C}D_0^{\alpha}g\right)(t) &= \left(I_a^{1-\alpha}{}'g\right)(t),\\ \left(I_a^{\alpha}g\right)(t) &= \frac{1}{\Gamma(\alpha)}\int_a^t \frac{g(\tau)}{(t-\tau)^{1-\alpha}}\,\mathrm{d}\tau, \quad t > a.\end{aligned} \tag{18}$$

For convenience, the following notations are used [17]:

$$\begin{aligned}\left({}_{(s)}^{C}D_a^{\alpha}g\right)(t) &= \frac{1}{\Gamma(1-\alpha)}\int_a^t \frac{g'(\tau)}{(s-\tau)^{\alpha}}\,\mathrm{d}\tau,\\ \left({}_{(s)}I_a^{\alpha}g\right)(t) &= \frac{1}{\Gamma(\alpha)}\int_a^t \frac{g(\tau)}{(s-\tau)^{1-\alpha}}\,\mathrm{d}\tau,\\ \left({}_{(s)}I_a^{\alpha}f\right)(t, w(t)) &= \frac{1}{\Gamma(\alpha)}\int_a^t \frac{f(\tau, w(\tau))}{(s-\tau)^{1-\alpha}}\,\mathrm{d}\tau.\end{aligned} \tag{19}$$

To obtain splitting schemes, the time-fractional differential system equation (13) is written in the integral form as follows:

$$w(t) = w(0) + A\left(I_0^{\alpha}w\right)(t) + \left(I_0^{\alpha}f\right)(t, w(t)), \quad 0 \le t \le T. \tag{20}$$

If one defines $t_n = nh$, $0 \le n \le N$, as uniformly distributed points on the range $[0,T]$ with $h = (T/N)$, then equation (20) is written over $(t_{n-1}, t_n]\,(n \ge 1)$ as follows:

$$\begin{aligned}w(t) = w(t_0) &+ I_{t_{n-1}}^{\alpha}Aw(t) + \left({}_{(t)}I_{t_0}^{\alpha}Aw\right)(t_{n-1})\\ &+ \left(I_{t_0}^{\alpha}f\right)(t, w(t)).\end{aligned} \tag{21}$$

In order to tailor a splitting scheme, the author of [17] defined

$$\begin{aligned}\overline{\overline{w}}(t) &= \widetilde{w}(t_{n-1}) + \left({}_{(t)}I_{t_0}^{\alpha}A\widetilde{w}\right)(t_{n-1}) + \left(I_{t_{n-1}}^{\alpha}A\overline{\overline{w}}\right)(t),\\ &t_{n-1} < t \le t_n,\end{aligned} \tag{22}$$

where the solution on $[0, t_{n-1}]$ is given and denoted with $\widetilde{w}(t)$ for $0 \le t \le t_{n-1}$.

The initial value of $\overline{\overline{w}}$ at t_{n-1} is defined as follows:

$$\widetilde{w}^{c}(t_{n-1}) = \widetilde{w}(t_{n-1}) + \left({}_{(t)}I_{t_0}^{\alpha}A\widetilde{w}\right)(t_{n-1}). \tag{23}$$

Then, equation (22) reads

$$\overline{\overline{w}}(t) = \widetilde{w}^{c}(t_{n-1}) + \left(I_{t_{n-1}}^{\alpha}A\overline{\overline{w}}\right)(t), \quad t_{n-1} < t \le t_n. \tag{24}$$

With equations (21) and (22), one has

$$\widetilde{w}(t) = \overline{\overline{w}}^{c}(t_n) + \left(I_{t_0}^{\alpha}f\right)(t, \widetilde{w}(t)), \quad t_{n-1} < t \le t_n, \tag{25}$$

where

$$\overline{\overline{w}}^{c}(t_n) = \overline{\overline{w}}(t) - \widetilde{w}(t_{n-1}) + w(t_0) - \left(I_{t_{n-1}}^{\alpha}A(\overline{\overline{w}} - \widetilde{w})\right)(t). \tag{26}$$

The solution $\tilde{w}(t)$ for $t \in (t_n, t_{n+1}]$ can be computed with repetitions of above processes.

Notice that from equations (23)–(26), one can reach equation (21). Thus, there is no splitting error.

With equations (23)–(26), one writes

$$\overline{\overline{w}}(t_n) = \tilde{w}^c(t_{n-1}) + \left(I^{\alpha}_{t_{n-1}} A\overline{\overline{w}}\right)(t_n), \tag{27}$$

$$\tilde{w}^c(t_{n-1}) = \tilde{w}(t_{n-1}) + \left({}_{(t_n)}I^{\alpha}_{t_0} A\tilde{w}\right)(t_{n-1}), \quad \tilde{w}_0 = w_0, \tag{28}$$

$$\tilde{w}(t_n) = \overline{\overline{w}}^c(t_n) + \left(I^{\alpha}_{t_0} f\right)(t_n, \tilde{w}(t_n)), \tag{29}$$

$$\overline{\overline{w}}^c(t_n) = \overline{\overline{w}}(t_n) - \tilde{w}(t_{n-1}) + w(t_0) - \left(I^{\alpha}_{t_{n-1}} A(\overline{\overline{w}} - \tilde{w})\right)(t_n). \tag{30}$$

The fully discrete time-splitting scheme can be tailored by discretization of equations (27)–(30). If one performs the weighted right-rectangle rule to equation (30), then the following is obtained

:

$$\begin{aligned}\overline{\overline{w}}^c(t_n) &\approx \overline{\overline{w}}(t_n) - \tilde{w}(t_{n-1}) + w(t_0) - \frac{1}{\Gamma(\alpha)}\int_{t_{n-1}}^{t_n} \frac{A\left(\overline{\overline{w}}(t_n) - \tilde{w}(t_n)\right)}{(t_n - \tau)^{1-\alpha}}\mathrm{d}\tau, \\ &= \overline{\overline{w}}(t_n) - \tilde{w}(t_{n-1}) + w(t_0) - \frac{Ah^{\alpha}}{\Gamma(1+\alpha)}\left(\overline{\overline{w}}(t_n) - \tilde{w}(t_n)\right).\end{aligned} \tag{31}$$

Using equation (31) with equation (29) yields

$$C^{\alpha}_h \tilde{w}_n = C^{\alpha}_h \overline{\overline{w}}_n + w_0 - \tilde{w}_{n-1} + \left(I^{\alpha}_{t_0} f\right)(t_n, \tilde{w}(t_n)), \tag{32}$$

where $\overline{\overline{w}}(t_n) = \overline{\overline{w}}_n$, $\tilde{w}(t_n) = \tilde{w}_n$, and $C^{\alpha}_h = I - A(h^{\alpha}/\Gamma(1+\alpha))$. If one defines all known information in equation (32) by $\overline{\overline{w}}^c_n$, then the following is obtained:

$$C^{\alpha}_h \tilde{w}_n = \overline{\overline{w}}^c_n + \left(I^{\alpha}_{t_{n-1}} f\right)(t_n, \tilde{w}(t_n)), \tag{33}$$

$$\overline{\overline{w}}^c_n = C^{\alpha}_h \overline{\overline{w}}_n + w_0 - \tilde{w}_{n-1} + \left({}_{(t_n)}I^{\alpha}_{t_0} f\right)(t_{n-1}, \tilde{w}(t_{n-1})). \tag{34}$$

On the other hand, the integrals in equations (28) and (34) are computed by performing the weighted trapezoidal rule and the weighted midpoint rule with $\tilde{w}(t_{j-(1/2)}) \approx (1/2)(\tilde{w}(t_{j-1}) - \tilde{w}(t_j))$, respectively. However, the integrals in equations (27) and (33) are computed by utilizing the weighted right-rectangle rule and the weighted left-rectangle rule, respectively. Then, the following time-splitting scheme is obtained [17]:

$$\begin{aligned}\overline{\overline{w}}_n &= \tilde{w}^c_{n-1} + \frac{Ah^{\alpha}}{\Gamma(1+\alpha)}\overline{\overline{w}}_n, \\ n &= 1, 2, 3, \ldots, N,\end{aligned} \tag{35}$$

$$\begin{aligned}\tilde{w}^c_{n-1} &= \tilde{w}_{n-1} + \frac{Ah^{\alpha}}{2\Gamma(1+\alpha)}\sum_{j=1}^{n-1} z^{\alpha}_{n,j}\left(\tilde{w}_{j-1} + \tilde{w}_j\right), \\ \tilde{w}_0 &= w_0,\end{aligned} \tag{36}$$

$$C^{\alpha}_h \tilde{w}_n = \overline{\overline{w}}^c_n + \frac{h^{\alpha}}{\Gamma(1+\alpha)} f\left(t_{n-1}, \tilde{w}_{n-1}\right), n = 1, 2, 3, \ldots, N, \tag{37}$$

$$\overline{\overline{w}}^c_n = C^{\alpha}_h \overline{\overline{w}}_n + w_0 - \tilde{w}_{n-1} + \frac{h^{\alpha}}{\Gamma(1+\alpha)}\sum_{j=1}^{n-1} z^{\alpha}_{n,j} f\left(t_{j-(1/2)}, \frac{\tilde{w}_{j-1} + \tilde{w}_j}{2}\right), \tag{38}$$

where

$$z_{n,j}^{\alpha} = \frac{\alpha}{h^{\alpha}} \int_{t_{j-1}}^{t_j} \frac{1}{(t_n - \tau)^{1-\alpha}} \mathrm{d}\tau = (n-j+1)^{\alpha} - (n-j)^{\alpha}. \tag{39}$$

The following approach is performed to get equation (36):

$$\begin{aligned} \left({}_{(t_n)}I_{t_0}^{\alpha} A\tilde{w}\right)(t_{n-1}) &= \frac{1}{\Gamma(\alpha)} \int_{t_0}^{t_{n-1}} \frac{A\tilde{w}(\tau)}{(t_n - \tau)^{1-\alpha}} \mathrm{d}\tau \\ &= \frac{1}{\Gamma(\alpha)} \sum_{j=1}^{n-1} \int_{t_{j-1}}^{t_j} \frac{A\tilde{w}(\tau)}{(t_n - \tau)^{1-\alpha}} \mathrm{d}\tau \\ &\approx \frac{A}{\Gamma(\alpha)} \sum_{j=1}^{n-1} \int_{t_{j-1}}^{t_j} \frac{(\tilde{w}_j + \tilde{w}_{j+1})}{2(t_n - \tau)^{1-\alpha}} \mathrm{d}\tau \\ &= \frac{Ah^{\alpha}}{2\Gamma(1+\alpha)} \sum_{j=1}^{n-1} z_{n,j}^{\alpha} (\tilde{w}_{j-1} + \tilde{w}_j). \end{aligned} \tag{40}$$

This time-splitting scheme is called as TS-I and the following convergence theorem is valid [17].

Theorem 1 (see [17]). *Let $w(t)$ and $\tilde{w}_n$, $n \in [0, N]$, be the solution of equation (13) and the schemes equations (35)–(38), respectively. Let $h = T/N$ and $Ah^{\alpha} \neq \Gamma(1+\alpha)I$.*

If $w \in C^2[0,T]$ and $f(t, w(t))$ fulfills the Lipschitz inequality equation (17), then following equation holds:

$$|w(t_n) - \tilde{w}_n| \leq K\left(1 + |w'(0)| t_n^{-\alpha}\right) h^{1+\alpha}, \quad 1 \leq n \leq N, \tag{41}$$

with a constants $K > 0$ that is independent of h. If $f \in C^2(G)$, then

$$|w(t_n) - \tilde{w}_n| \leq K\left(1 + t_n^{\alpha-1}\right) h^{1+\alpha}, \quad 1 \leq n \leq N. \tag{42}$$

4. Numerical Results

To compare the performance of the mentioned algorithms for each test examples, the following absolute error and L_2 and L_{∞} error norms are used:

$$\begin{aligned} \left\|u^{\mathrm{num}} - u\right\|_{L_2} &= \sqrt{\frac{1}{k} \sum_{j=0}^{k} \left(u_{j,N} - u(x_j, t)\right)^2}, \\ \left\|u^{\mathrm{num}} - u\right\|_{L_{\infty}} &= \max_{0 \leq j \leq k} \left|u_{j,N} - u(x_j, t)\right|, \\ \mathrm{Abs.Error} &= \left|u_{j,N} - u(x_j, t)\right|, \quad 0 \leq j \leq k. \end{aligned} \tag{43}$$

We consider to assign the shape constant using trial and error. We have used errors between exact and numerical solutions to determine the best value of shape parameters. As shown in Figure 1, the selection of its best value $c = 0.72$ with corresponding smallest error. We compute L_{∞} error norm between exact and numerical solutions at the interval [0.01, 10] and compare them to specify the shape parameter.

Example 1. We first compute the approximate solutions of problem (1) satisfying the conditions as follows:

$$\begin{aligned} &u(x, t=0) = 0, \quad t \geq 0, \\ &u(x=0, t) = t^2, \\ &u(x=1, t) \\ &\qquad = -t^2, \quad 0 \leq x \leq 1, \end{aligned} \tag{44}$$

where

$$g(x,t) = \left(\frac{2t^{2-\alpha}}{\Gamma(3-\alpha)} - \pi^4 t^4 \sin(\pi x) + \nu\pi^2 t^2\right)\cos(\pi x). \tag{45}$$

The exact solution is [19]

$$u(x,t) = t^2 \cos(\pi x). \tag{46}$$

A comparison of the errors between presented method and [19, 20] for Example 1 is given in Table 1 at $T = 0.1$ with $h = 0.0005$. The results are obtained with $k = 10$ and $\nu = 1$, $\nu = 0.5$, and $\nu = 0.1$ with corresponding shape parameter values $c = 0.7887$, $c = 5.71$, and $c = 0.72$, respectively, for the present method. It is clear that as the viscosity value ν decreases, the errors become smaller as well as better than [19, 20]. In Table 2, we compare the numerical results with those of [31] for Example 1 at $T = 1$ with $h = 0.02$. For the presented scheme, the computation was handled using $c = 0.05$, $c = 0.0517$, $c = 0.0536$ for $\alpha = 0.25$, $\alpha = 0.75$, $\alpha = 0.9$, respectively, and $k = 10$. It indicates that the results obtained with the presented algorithm are in good match with the exact solutions and absolute errors are better than those given in [31]. We have tabulated the numerical solutions for Example 1 at $T = 1$ with $h = 0.01$, $\nu = 0.1$, and $\alpha = 0.9$ in Table 3. The shape parameters are $c = 3.933$, $c = 5.138$, and $c = 5.289$ for $k = 30$, $k = 50$,and $k = 70$, respectively. It shows that the approximate solutions are in good match with exact solutions and the absolute errors are small enough. Furthermore, Figures 2 and 3 show the behavior of the approximate solutions at $T = 0.1$ and the absolute errors, respectively, with $h = 0.0005$, $c = 0.72$, $\nu = 0.1$, $k = 10$, and $\alpha = 0.5$. Figure 2 shows that the obtained solutions are symmetric.

Example 2. In the next example, we deal with the time-fractional Burgers' equation (1) subject to following condition:

$$u(x, t=0) = \frac{1}{-1 + 5(\cosh(x/2) - \sinh(x/2))}, \quad t \geq 0. \tag{47}$$

The exact solution is given as [32]

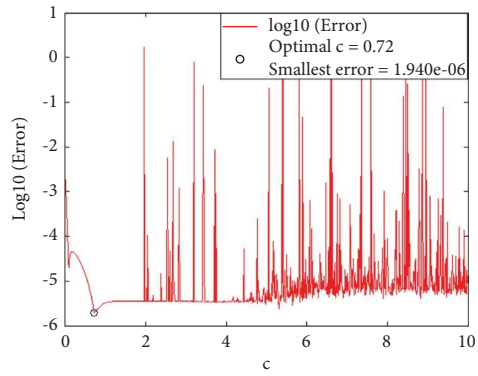

Figure 1: The best value of shape parameter c for Example 1 at $T = 0.1$ with $h = 0.0005$, $k = 10$, and $\nu = 0.1$.

Table 1: Error norms of Example 1 at $T = 0.1$ for $h = 0.0005$ and $\alpha = 0.5$.

	$\nu = 1$		$\nu = 0.5$		$\nu = 0.1$	
	L_2	L_∞	L_2	L_∞	L_2	L_∞
Reference [19] ($m' = 10$)	$6.980e-06$	$9.547e-06$	$6.492e-06$	$8.854e-06$	$4.288e-06$	$5.714e-06$
Reference [20] ($k = 80$)	$6.528e-06$	$9.164e-06$	$5.835e-06$	$8.250e-06$	$3.105e-06$	$4.847e-06$
Present ($k = 10$)	$3.601e-05$	$6.175e-05$	$1.386e-06$	$2.410e-06$	$1.428e-06$	$1.940e-06$

Table 2: Results of Example 1 at $T = 1$ for $h = 0.02$.

α	x	[31] $k = 50$	Present $k = 10$	Exact	Abs. error [31]	Abs. error Present
0.25	0.10	0.9812176	0.9579127	0.9510565	$3.016115e-02$	$6.856230e-03$
	0.30	0.6464059	0.5680175	0.5877853	$5.862068e-02$	$1.976778e-02$
	0.50	$-3.075e-59$	$-4.5661e-16$	$6.1232e-17$	$3.304634e-59$	$5.178393e-16$
	0.70	−0.664067	−0.5680175	−0.5877853	$5.862068e-02$	$1.976778e-02$
	0.90	−9.812136	−0.9579127	−0.9510565	$3.016115e-02$	$6.856230e-03$
0.75	0.10	0.9319893	0.9643378	0.9510565	$1.906739e-02$	$1.328133e-02$
	0.30	0.5418891	0.5746742	0.5877853	$4.589585e-02$	$1.311105e-02$
	0.50	$6.120e-59$	$-6.1287e-15$	$6.1232e-17$	$5.890384e-59$	$6.189963e-15$
	0.70	−0.541856	−0.5746742	−0.5877853	$4.589585e-02$	$1.311105e-02$
	0.90	−0.931992	−0.9643378	−0.9510565	$1.906739e-02$	$1.328133e-02$
0.9	0.10	0.9146401	0.9628647	0.9510565	$3.641245e-02$	$1.180823e-02$
	0.30	0.5061482	0.5757488	0.5877853	$8.163693e-02$	$1.203643e-02$
	0.50	$-1.950e-59$	$1.9096e-15$	$6.1232e-17$	$2.179615e-59$	$1.848389e-15$
	0.70	−0.506129	−0.5757488	−0.5877853	$8.163693e-02$	$1.203643e-02$
	0.90	−0.914626	−0.9628647	−0.9510565	$3.641245e-02$	$1.180823e-02$

$$u(x,t) = \frac{1}{-1 + 5\left(\cosh\left((x/2) + \left(t^{\alpha}/4\Gamma(1+\alpha)\right)\right) - \sinh\left((x/2) + \left(t^{\alpha}/4\Gamma(1+\alpha)\right)\right)\right)}, \tag{48}$$

with force term

$$g(x,t) = 0. \tag{49}$$

The boundary conditions are computed by using the exact solution.

In Table 4, numerical solutions and absolute errors at $T = 0.02$ are obtained by the present method for Example 2 with $c = 3.88$, $k = 50$, $\alpha = 0.8$, and $h = 0.0001$. Table 4 shows that the numerical solutions of the presented methods are highly accurate than the solutions tabulated in [32] with the same parameter values.

Table 3: Results of Example 1 at $T = 1$, $h = 0.01$, $\nu = 0.1$, and $\alpha = 0.9$.

k	x	Present	Exact	Abs. error
30	0.10	0.9510383	0.9510565	$1.823706e-05$
	0.30	0.5877661	0.5877853	$1.917656e-05$
	0.70	−0.5877274	−0.5877853	$5.787549e-05$
	0.90	−0.9510221	−0.9510565	$3.439243e-05$
50	0.10	0.9510147	0.9510565	$4.184683e-05$
	0.30	0.5877117	0.5877853	$7.358647e-05$
	0.70	−0.5877305	−0.5877853	$5.470573e-05$
	0.90	−0.9509895	−0.9510565	$6.705913e-05$
70	0.10	0.9510408	0.9510565	$1.568811e-05$
	0.30	0.5877369	0.5877853	$4.836005e-05$
	0.70	−0.5877776	−0.5877853	$7.668952e-06$
	0.90	−0.9510778	−0.9510565	$2.125912e-05$

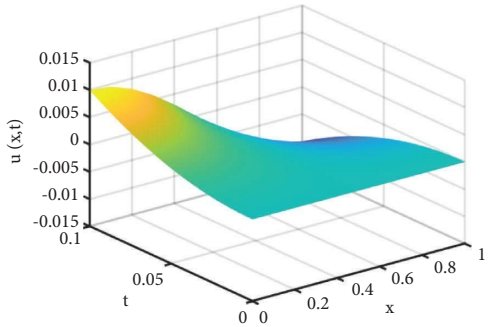

Figure 2: Numerical solutions of Example 1 at $T = 0.1$ with $h = 0.0005$, $c = 0.72$, $\nu = 0.1$, $k = 10$, and $\alpha = 0.5$.

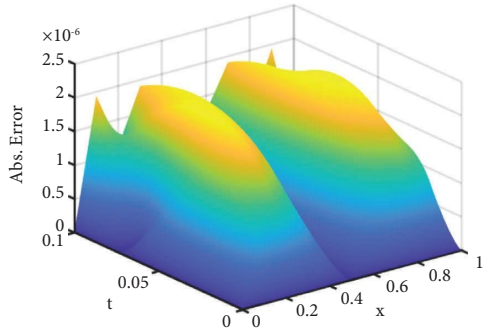

Figure 3: Absolute errors of Example 1 at $T = 0.1$ with $h = 0.0005$, $c = 0.72$, $\nu = 0.1$, $k = 10$, and $\alpha = 0.5$.

Example 3. As a final example, we investigate the numerical solutions of (1) satisfying the conditions as follows:

$$\begin{aligned} &u(x, t = 0) = \sin(\pi x), \\ &u(x = 0, t) = u(x = 1, t) = 0, \quad t \geq 0. \end{aligned} \tag{50}$$

The series solutions are obtained in [33] by performing the Hopf-Cole technique [34, 35] as follows:

$$u(x,t) = 2\nu\pi \frac{\sum_{m=1}^{\infty} b_m e^{-m^2\pi^2\nu t^{\alpha}} m \sin(m\pi x)}{b_0 + \sum_{m=1}^{\infty} b_m e^{-m^2\pi^2\nu t^{\alpha}} \cos(m\pi x)}, \tag{51}$$

where

$$\begin{aligned} b_0 &= \int_0^1 e^{-(2\pi\nu)^{-1}[1-\cos(\pi x)]} dx, \\ b_m &= 2\int_0^1 e^{-(2\pi\nu)^{-1}[1-\cos(\pi x)]} \cos(m\pi x) dx, \\ &(m = 1, 2, 3 \ldots). \end{aligned} \tag{52}$$

In Table 5, numerical solutions and absolute errors at $T = 0.1$ are computed by the present scheme for Example 3 with $k = 10$ and $\alpha = 0.9$. The shape parameters are chosen as $c = 0.76$, $c = 0.05$, and $c = 0.06$ for $h = 0.01$, $h = 0.001$, and $h = 0.0001$, respectively. Table 5 shows a good match between the obtained and the analytical results.

Table 4: Comparison of approximate solutions and absolute errors of Example 2 with [32] at $T = 0.02$ for $h = 0.0001$, $\nu = 1$, and $\alpha = 0.8$.

x	[32]	Present	Exact	Abs. error [32]	Abs. error
0.02	0.256321	0.2569083	0.2569058	$5.84566e-04$	$2.49983e-06$
0.04	0.259566	0.2601621	0.2601595	$5.93809e-04$	$2.62925e-06$
0.06	0.262860	0.2634659	0.2634630	$6.03243e-04$	$2.96258e-06$
0.08	0.266204	0.2668207	0.2668173	$6.12874e-04$	$3.42370e-06$
0.10	0.269601	0.2702275	0.2702235	$6.22707e-04$	$3.96545e-06$
0.12	0.273050	0.2736871	0.2736825	$6.32749e-04$	$4.53857e-06$

Table 5: Approximate solutions and absolute errors of Example 3 at $T = 0.1$, for values of $\nu = 1$ and $\alpha = 0.9$.

h	x	Present	Exact	Abs. error
0.01	0.10	0.1010647	0.0747999	$2.62648e-02$
	0.30	0.2387890	0.1984250	$4.03640e-02$
	0.70	0.2377580	0.2073367	$3.04213e-02$
	0.90	0.0404055	0.0803122	$3.99067e-02$
0.001	0.10	0.0802966	0.0747999	$5.49670e-03$
	0.30	0.1982625	0.1984250	$1.62500e-04$
	0.70	0.2040926	0.2073367	$3.24410e-03$
	0.90	0.0842372	0.0803122	$3.92490e-03$
0.0001	0.10	0.0713870	0.0747999	$3.41300e-03$
	0.30	0.2004805	0.1984250	$2.05560e-03$
	0.70	0.2070095	0.2073367	$3.27100e-04$
	0.90	0.0751544	0.0803122	$5.15780e-03$

5. Conclusions

A numerical algorithm based on the combination of the MQ-RBF and splitting technique is presented to investigate the solutions of the fractional Burgers' equation, numerically. Instead of classical splitting methods, a division method for the fractional nonlinear systems of ordinary differential equations is performed. This method has $O(h^{1+\alpha})$ order of convergence which is different from $O(h^{2\alpha})$. That is, when α tends to zero, the error does not increase significantly. Furthermore, this algorithm also takes advantage of a meshfree scheme. It is concluded that the present approach yields accurate results and exhibits physical dynamics of the time-fractional Burgers' equation. This proposed scheme can be used for the other nonlinear fractional PDEs.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

The authors completed this study and wrote and approved the final version of the manuscript.

Acknowledgments

The authors present their deepest thanks to Dr. Muaz Seydaoglu for his guidance on simulation and his careful reading of the revised version of the paper.

References

[1] H. Bateman, "Some recent researchers on the motion of fluids," *Monthly Weather Review*, vol. 43, pp. 171–199, 1948.
[2] J. M. Burger, *A Mathematical Model Illustrating the Theory of Turbulence*, Academic Press, Cambridge, MA, USA, 1948.
[3] A. Kilbas, H. Srivastava, and J. Trujillo, *Theory and Applications of Fractional Differential Equations*, Elsevier, Amsterdam, Netherlands, 2006.
[4] I. Podlubny, *Fractional Differential Equations of Mathematics in Science and Engineering*, Academic Press, Cambridge, MA, USA, 1999.
[5] K. Diethelm, N. J. Ford, and A. D. Freed, "Detailed error analysis for a fractional Adams method," *Numerical Algorithms*, vol. 36, no. 1, pp. 31–52, 2004.
[6] Z. Z. Sun and X. Wu, "A fully discrete difference scheme for a diffusion-wave system," *Applied Numerical Mathematics*, vol. 56, no. 2, pp. 193–209, 2006.
[7] K. Diethelm, "Smoothness properties of solutions of Caputo-type fractional differential equations," *Fractional Calculus and Applied Analysis*, vol. 10, pp. 151–160, 2007.
[8] W. Deng, "Short memory principle and a predictor–corrector approach for fractional differential equations," *Journal of Computational and Applied Mathematics*, vol. 206, no. 1, pp. 174–188, 2007.
[9] L. Galeone and R. Garrappa, "Explicit methods for fractional differential equations and their stability properties," *Journal of*

Computational and Applied Mathematics, vol. 228, no. 2, pp. 548–560, 2009.

[10] M. Cui, "Compact finite difference method for the fractional diffusion equation," *Journal of Computational Physics*, vol. 228, no. 20, pp. 7792–7804, 2009.

[11] F. Zeng, C. Li, F. Liu, and I. Turner, "The use of finite difference/element approaches for solving the time-fractional subdiffusion equation," *SIAM Journal on Scientific Computing*, vol. 35, no. 6, pp. A2976–A3000, 2013.

[12] C. Li and F. Zeng, "The finite difference methods for fractional ordinary differential equations," *Numerical Functional Analysis and Optimization*, vol. 34, no. 2, pp. 149–179, 2013.

[13] B. Zhou, X. Chen, and D. Li, "Nonuniform Alikhanov linearized Galerkin finite element methods for nonlinear time-fractional parabolic equations," *Journal of Scientific Computing*, vol. 85, no. 2, pp. 39–20, 2020.

[14] D. Li, C. Zhang, and M. Ran, "A linear finite difference scheme for generalized time fractional Burgers equation," *Applied Mathematical Modelling*, vol. 40, no. 11-12, pp. 6069–6081, 2016.

[15] M. Sari, H. Tunc, and M. Seydaoglu, "Higher order splitting approaches in analysis of the Burgers equation," *Kuwait Journal of Science*, vol. 46, no. 1, 2019.

[16] M. Seydaoglu, Y. Uçar, and S. Kutluay, "An efficient Strang splitting technique combined with the multiquadric-radial basis function for the Burgers' equation," *Computational and Applied Mathematics*, vol. 40, no. 8, pp. 309–328, 2021.

[17] W. Cao, Z. Zhang, and G. E. Karniadakis, "Time-splitting schemes for fractional differential equations I: smooth solutions," *SIAM Journal on Scientific Computing*, vol. 37, no. 4, pp. A1752–A1776, 2015.

[18] A. Esen and O. Tasbozan, "Numerical solution of time fractional Burgers' equation," *Acta Universitatis Sapientiae, Mathematica*, vol. 7, no. 2, pp. 167–185, 2015.

[19] O. Oruc, A. Esen, and F. Bulut, "A unified finite difference Chebyshev wavelet method for numerically solving time fractional Burgers' equation," *Discrete and Continuous Dynamical Systems*, vol. 12, no. 3, pp. 533–542, 2019.

[20] A. Esen and O. Tasbozan, "Numerical solution of time fractional Burgers' equation by cubic B-spline finite elements," *Mediterranean Journal of Mathematics*, vol. 13, no. 3, pp. 1325–1337, 2016.

[21] V. Kumar, S. Singh, and M. E. Koksal, "A composite algorithm for numerical solutions of two-dimensional coupled Burgers' equations," *Journal of Mathematics*, vol. 2021, pp. 1–13, 2021.

[22] L. Li and D. Li, "Exact solutions and numerical study of time fractional Burgers' equations," *Applied Mathematics Letters*, vol. 100, Article ID 106011, 2020.

[23] M. Uddin, H. U. Jan, and M. Usman, "RBF-PS method for approximation and eventual periodicity of fractional and integer type KdV equations," *Partial Differential Equations in Applied Mathematics*, 2022.

[24] G. Hariharan, G. Swaminathan, and B. Sripathy, "An efficient operational matrix approach for the solutions of Burgers' and fractional Burgers' equations using wavelets," *Journal of Mathematical Chemistry*, vol. 59, no. 2, pp. 554–573, 2021.

[25] A. Esen, F. Bulut, and Ö. Oruç, "A unified approach for the numerical solution of time fractional Burgers' type equations," *The European Physical Journal Plus*, vol. 131, no. 4, pp. 116–213, 2016.

[26] M. Onal and A. Esen, "A Crank-Nicolson approximation for the time fractional Burgers equation," *Applied Mathematics and Nonlinear Sciences*, vol. 5, no. 2, pp. 177–184, 2020.

[27] L. Zada and I. Aziz, "The numerical solution of fractional Korteweg-de Vries and Burgers' equations via Haar wavelet," *Mathematical Methods in the Applied Sciences*, vol. 44, no. 13, pp. 10564–10577, 2021.

[28] E. J. Kansa, "Multiquadrics—a scattered data approximation scheme with applications to computational fluid-dynamics—I surface approximations and partial derivative estimates," *Computers and Mathematics With Applications*, vol. 19, no. 8-9, pp. 127–145, 1990.

[29] E. J. Kansa, "Multiquadrics—a scattered data approximation scheme with applications to computational fluid-dynamics—II solutions to parabolic, hyperbolic and elliptic partial differential equations," *Computers and Mathematics with Applications*, vol. 19, no. 8-9, pp. 147–161, 1990.

[30] S. Blanes and F. Casas, *A Concise Introduction to Geometric Numerical Integration*, CRC Press, Boca Raton, FL, USA, 2016.

[31] M. S. Hashmi, M. Wajiha, S. W. Yao, A. Ghaffar, and M. Inc, "Cubic spline based differential quadrature method: a numerical approach for fractional Burger equation," *Results in Physics*, vol. 26, Article ID 104415, 2021.

[32] A. Yokus and D. Kaya, "Numerical and exact solutions for time fractional Burgers' equation," *The Journal of Nonlinear Science and Applications*, vol. 10, no. 07, pp. 3419–3428, 2017.

[33] A. Kurt, Y. Cenesiz, and O. Tasbozan, "On the solution of Burgers' equation with the new fractional derivative," *Open Physics*, vol. 13, no. 1, 2015.

[34] E. Hopf, "The partial differential equation $U_t+UU_x=\mu U_{xx}$," *Communications on Pure and Applied Mathematics*, vol. 3, pp. 201–230, 1950.

[35] J. D. Cole, "On a quasi linear parabolic equation occurring in aerodynamics," *Quarterly of Applied Mathematics*, vol. 9, no. 3, pp. 225–236, 1951.