Results in Physics 26 (2021) 104415

Contents lists available at ScienceDirect

Results in Physics

journal homepage: www.elsevier.com/locate/rinp

Cubic spline based differential quadrature method: A numerical approach for fractional Burger equation

Muhammad Sadiq Hashmi [a], Misbah Wajiha [a], Shao-Wen Yao [b,*], Abdul Ghaffar [c], Mustafa Inc [d,e,f,*]

[a] Department of Mathematics, The Government Sadiq College Women University, Bahawalpur 63100, Pakistan
[b] School of Mathematics and Information Science, Henan Polytechnic University, Jiaozuo 454000, China
[c] Department of Mathematics, Ghazi University, D G Khan 32200, Pakistan
[d] Department of Computer Engineering, Biruni University, Istanbul, Turkey
[e] Department of Mathematics, Science Faculty, Firat University, 23119 Elazig, Turkey
[f] Department of Medical Research, China Medical University Hospital, China Medical University, Taichung, Taiwan

ARTICLE INFO

Keywords:
Differential Quadrature Method
Stability analysis
Burger equation

ABSTRACT

In this research paper, our main objective is to represent a direct numerical approach for solving time-fractional Burger's equation using modified hybrid B-spline basis function. The Caputo derivative is used to discretize the time-fractional derivative and for Space derivative Differential Quadrature Method (DQM) based on B-Spline is used. The DQM method has its own inherited advantage being a simple and programable method. The embedding of B-spline basis makes it more practical to approximate the solution curve. DQM with B-spline basis is a simple and efficient technique based on the matrix approach. The problem is discretized in the system of non-linear equations and then further solved by a programming tools. The stability is examined by the matrix-based approach. The presented method has been applied to three test problems. The obtained results showed that the proposed method is good for solving non-linear time-fractional Burger's equation. The approximated solutions are graphically represented and the results showed that solutions are closed to the exact solution.

Introduction

Burger's equation was first introduced by Batemann [1], after that; this equation is analyzed by Burgers [2]. Burger's equation is used as a model in many of the fields like in disperse water Whitham [3] and in heat conduction Bluman et al. [4] etc. The general form of the equation is

$$\frac{\partial^\alpha u(\varrho,\tau)}{\partial\tau^\alpha}+u\left(\varrho,\tau\right)\frac{\partial u(\varrho,\tau)}{\partial\varrho}-\mu\frac{\partial^2 u(\varrho,\tau)}{\partial\varrho^2}=0,\quad u\left(\varrho,0\right)=\phi\left(\varrho\right),\quad a\leqslant\varrho\leqslant b, \tag{1.1}$$

and the boundary conditions are

$$u(a,\tau)=\phi_1(\tau),\qquad u(b,\tau)=\phi_2(\tau),\quad 0<\alpha\leqslant 1. \tag{1.2}$$

Here $\phi(\varrho),\phi_1(\tau),\phi_2(\tau)$ are smooth functions and ϱ,τ are space and time variables respectively.

Several numerical techniques have been used for solving fractional partial differential equations such as Adomain decomposition by Ford and Conolly [5] and in order to obtain the solution for non-linear partial differential equation, Dehghan et al. [6] proposed variational iteration method and Mokhtary [7] used Legendre collocation method. The finite difference scheme for a time-fractional diffusion equation was proposed by Lin and Xu [8]. Recently, the differential quadrature method has gained popularity in the research field. Bellman et al. [9] introduced DQ method, which is broadly used in numerical solutions of PDEs. The differential quadrature method is a higher-order numerical discretization approach that provides very accurate solutions to differential equations with tremendously less computational attempts. This method is explicit, easy to use and especially implemented for non-linear cases. The partial derivative of an unknown function in a particular direction as a weighted sum of functional values at all discrete points can be approximated by DQM. The main idea of this method is that the weighted coefficients are only dependent on grid spacing and

* Corresponding authors at: School of Mathematics and Information Science, Henan Polytechnic University, Jiaozuo 454000, China (S.-W. Yao). Department of Computer Engineering, Biruni University, Istanbul, Turkey (M. Inc).
E-mail addresses: yaoshaowen@hpu.edu.cn (S.-W. Yao), minc@firat.edu.tr (M. Inc).

https://doi.org/10.1016/j.rinp.2021.104415
Received 14 April 2021; Received in revised form 30 May 2021; Accepted 31 May 2021
Available online 10 June 2021

independent of any specific problem. In DQM with the efficient use of B-spline as a basis function, we can tackle linear and non-linear initial/boundary value problems. Abdul Majeed et al. [10] solved the time-fractional Burger and Fisher equation using cubic B-spline approximation method. Mittal et al. [11] proposed numerical simulation on hyperbolic diffusion equations using modified cubic B-spline DQM. Dahiya et al. [12] have proposed numerical solutions of differential equations using modified B-spline DQM. Yaseen et al. [13] gives an efficient computational technique based on cubic trigonometric B-splines for time-fractional Burger's equation. Agarwal [14,15] gives a general solution for the fourth-order fractional diffusion equation, and also gives it's a general solution that is defined in a bounded domain. Esen et al. [16] gives a numerical solution of time-fractional Burger's equation. Shukla et al. [17] presented the numerical solution of two dimensional coupled viscous Burger equation using modified cubic B-spline DQM. Chun-Ku Kuo et al. [18] represent a new exact solution of Burger's equation with the linearized solution. Yousif et al. [19] show the solution of moving boundary space–time fractional Burger's equation.

The analytical and numerical solution of fractional fuzzy hybrid system in Hilbert space are calculated by Shatha Hassan et al. [20]. Under conformable fractional derivative sense Omer Abu et al. [21] gives numerical solution of coupled fractional resonant Schrodinger equations arising in quantum mechanics. Al-Samadi et al. [22] used computational algorithm for solving Fredholm time-fractional partial integro-differential equations of dirichlet functions type with error estimates. Samir at el. [23] used conformable residual power series method to compute approximate solutions of nonlinear fractional Kundu-Eckhaus and coupled fractional massive Thirring equations. Jiwari et al. [24] develop a mesh free algorithm that is based on radial basis function and Differential Quadrature technique to calculate the shock behaviour of Burger equation. Vineet et al. [25] proposed an algorithm that is based on exponential modified cubic B-spline differential quadrature method for non-linear Burger equation. The numerical solution of Burger equation by Hybrid numerical scheme is done by Jiwari [26]. Sharma et al. [27] gives numerical solution of Burger equation that is based on weighted average differential quadrature method. Mital et al. [28] gives differential quadrature method for Burger type equation. Hashmi et al. [29] solved Hunter Saxton equation using B-spline method. Recently, Rabia et el. [30] has efficiently applied B-spline collocation method for the solution of space fractional PDEs.

The simplicity and applicability of DQM and the efficiency of B-spline basis function to approximate the solution curve has motivated us to use a B-spline based DQM method using trigonometric and polynomial basis as a test function for the solution of Burger equation. To the best of my knowledge the current attempt inherits novality as no attempt has been seen in literature to solve first-order fractional PDE using the B-spline based DQM. Here the fractional derivative for $0<\alpha<1$, in the form of term involving α derivative, are discretized using caputo derivative. The space derivative is approximated using the DQ technique based on the modified B-spline. The hybrid version of the polynomial and trigonometric spline is used to represent the output in both versions. In order to prove stability, the matrix-based approach is used which adopts the induction approach to prove the required result, various numerical examples are been incorporated to validate the procedure.

The outline of the paper is as following: Section "Preliminaries" contains some preliminaries which help to understand the paper. Section "Description of proposed method" describes the description of method and the procedure of calculating the weighted coefficients and construction of the algorithm is given in Section "Calculation of weighted coefficients". Stability analysis is presented in Section "Stability analysis". Numerical examples and conclusion is presented in Section "Numerical experiments" and "Concluding remarks" respectively.

Preliminaries

Here, are some basic preliminaries to understand the concepts. Let M, $N\in\mathbb{Z}^+$ and a grid points have the form, involved in paper.

$$\Omega_{d\tau}=(\tau_n:\tau_n=d\tau,0\leqslant n\leqslant N),$$

$$\Omega_{\hbar}=(\varrho_i:\varrho_i=a+i\hbar,0\leqslant i\leqslant M),$$

with $\tau=\frac{d\tau}{N}$, $\hbar=\frac{b-a}{M}$ on $(0,d\tau]\times[a,b]$, then some results are introduced as follows:

Fractional derivative

The α^{th} order Caputo and Reimann Liouville derivative are given as: ([31,32]).

$$ {}_0^C D_\tau^\alpha g\left(\varrho,\tau\right)=\frac{1}{\Gamma(m-a)}\int_0^\tau\frac{\partial^m g(\varrho,\xi)}{\partial\xi^m}\frac{d\xi}{(\tau-\xi)^{1+\alpha-m}}, \tag{2.1}$$

$$ {}_0^{RL} D_\tau^\alpha g\left(\varrho,\tau\right)=\frac{1}{\Gamma(m-a)}\frac{\partial^m}{\partial\tau^m}\int_0^\tau\frac{g(\varrho,\xi)d\xi}{(\tau-\xi)^{1+\alpha-m}}, \tag{2.2}$$

where, $m-1<\alpha<m,\quad m\in\mathbb{Z}^+$.

Discritized form

Two frequently used fractional derivative i.e. Caputo and Reimann-Lioullive derivatives become equal with the existence of an additive factor as

$$ {}_0^C D_\tau^\alpha g\left(\varrho,\tau\right)={}_0^{RL} D_\tau^\alpha g\left(\varrho,\tau\right)-\sum_{l=0}^{n-1}\frac{g^l(\varrho,0)\tau^{l-\alpha}}{\Gamma(l-\alpha+1)}. \tag{2.3}$$

and for the description of Reimann-Liouville use a proper scheme on right hand side of (2.3), difference scheme for Caputo derivative can be written as:

$$ D_\tau^\alpha f\left(\varrho,\tau_n\right)\cong\frac{1}{d\tau^\alpha}\sum_{k=0}^{n}w_k^\alpha f\left(\varrho,\tau_{n-k}\right)-\frac{1}{d\tau^\alpha}\sum_{z=0}^{m-1}\sum_{k=0}^{n}\frac{w_k^\alpha f^z(\varrho,0)\tau_{n-k}^z}{z!}, \tag{2.4}$$

where the coefficient of the discrete form is [33]

$$ w_k^\alpha=(-1)^k\binom{\alpha}{k}=\frac{\alpha(k-\alpha)}{\Gamma(-\alpha)\Gamma(k+1)}; \tag{2.5}$$

which satisfy the following properties

- $w_0^\alpha=1,\qquad w_k^\alpha<0\quad\forall\, k\geqslant1,$
- $\sum_{k=0}^\infty w_k^\alpha=0\,,\qquad\sum_{k=0}^{n-1}w_k^\alpha>0.$

When $0<\alpha<1$,the above equation becomes

$$ D_\tau^\alpha f\left(\varrho,\tau_n\right)=\frac{1}{d\tau^\alpha}\sum_{k=0}^{n}w_k^\alpha f\left(\varrho,\tau_{n-k}\right)-\frac{1}{d\tau^\alpha}\sum_{z=0}^{m-1}w_k^\alpha f\left(\varrho,s\right)+R_\tau. \tag{2.6}$$

Table 1
Absolute errors of Problem 1 for $\alpha = 0.5, M = 50, N = 50$ and $d\tau = 0.01$.

ϱ	$g = 0$	$g = 0.5$	$g = 1$
0.10	$9.301531e^{-5}$	$9.717916e^{-4}$	$2.7874955e^{-4}$
0.30	$2.135400e^{-4}$	$1.333536e^{-3}$	$1.4433254e^{-3}$
0.50	$2.262922e^{-4}$	$7.572175e^{-4}$	$1.2128387e^{-3}$
0.70	$1.515313e^{-4}$	$5.366826e^{-4}$	$4.2563002e^{-4}$
0.90	$9.607724e^{-5}$	$8.111253e^{-4}$	$3.057444e^{-4}$

Table 2
Numerical displays of absolute errors of Problem 1 at $\alpha = 0.5, M = 40$ and $N = 50$.

τ	ϱ	Exact	MHB-DQM	Error
0.1	0.10	0.014356	0.01437327	$1.313385e^{-5}$
	0.30	0.031233	0.03127053	$3.670953e^{-5}$
	0.50	0.022360	0.02239636	$3.558464e^{-5}$
	0.70	-0.00494	-0.0049262	$2.020413e^{-5}$
	0.90	-0.02817	-0.0281678	$1.113439e^{-5}$
0.01	0.10	$4.5399e^{-4}$	$4.551443e^{-4}$	$1.156137e^{-6}$
	0.30	$9.8768e^{-4}$	$9.902324e^{-4}$	$2.544462e^{-6}$
	0.50	$7.0710e^{-4}$	$7.709302e^{-4}$	$2.195274e^{-6}$
	0.70	$-1.5643e^{-4}$	$-1.55685e^{-4}$	$7.487593e^{-7}$
	0.90	$-8.9100e^{-4}$	$-8.90991e^{-4}$	$1.485272e^{-8}$

Table 3
Numerical displays of errors of Problem 1 at $\alpha = 0.9, M = 40$ and $N = 50$.

τ	ϱ	Exact	MHB-DQM	Error
0.1	0.10	0.160509	0.16067301	$1.632043e^{-4}$
	0.30	0.349200	0.34957271	$3.722197e^{-4}$
	0.50	0.250000	0.25039461	$3.944911e^{-4}$
	0.70	-0.05530	-0.0550438	$2.623599e^{-4}$
	0.90	-0.31501	-0.3148649	$1.569171e^{-4}$
0.01	0.10	$4.53990e^{-4}$	$4.586741e^{-4}$	$1.684234e^{-6}$
	0.30	$9.87688e^{-4}$	$9.978392e^{-4}$	$1.015145e^{-6}$
	0.50	$7.07106e^{-4}$	$7.144183e^{-4}$	$7.311678e^{-6}$
	0.70	$-1.56434e^{-4}$	$-1.57379e^{-4}$	$9.452739e^{-7}$
	0.90	$-8.91006e^{-4}$	$-8.95148e^{-4}$	$4.141547e^{-6}$

Table 4
Comparison of numerical and exact solutions of Problem 1 at different values of $\tau, \alpha = 0.5, M = 50$ and $N = 100$.

τ	ϱ	Exact	MHB-DQM	Error
0.5	0.10	0.160509	0.160564	$5.507027e^{-5}$
	0.30	0.349200	0.349323	$1.230118e^{-4}$
	0.50	0.250000	0.250138	$1.389700e^{-4}$
	0.70	-0.0221	-0.02208	$1.124788e^{-4}$
	0.90	-0.31501	-0.31492	$9.618957e^{-5}$
0.75	0.10	0.294875	0.294964	$8.886032e^{-5}$
	0.30	0.641522	0.641725	$2.027605e^{-4}$
	0.50	0.459279	0.459523	$2.446133e^{-4}$
	0.70	-0.10160	-0.10139	$2.140077e^{-4}$
	0.90	-0.57872	-0.57853	$1.863363e^{-4}$
1	0.10	0.453990	0.454115	$1.248275e^{-4}$
	0.30	0.987688	0.987981	$2.930343e^{-4}$
	0.50	0.707106	0.707484	$3.778662e^{-4}$
	0.70	-0.15643	-0.15608	$3.509335e^{-4}$
	0.90	-0.89100	-0.89070	$3.025356e^{-4}$

Cubic B-spline basis

Let $\varrho_{-i} = a - i\hbar, \varrho_{M+i} = b + i\hbar$ and $i = 1, 2, 3$ be the six knots outside the interval. The cubic trigonometric B-spline is defined as
where,

$$q\left(\varrho_m\right) = \sin\left(\frac{\varrho - \varrho_m}{2}\right), \quad p\left(\varrho_m\right) = \sin\left(\frac{\varrho_m - \varrho}{2}\right)$$

and $\delta = \sin\left(\frac{\hbar}{2}\right)\sin\left(\hbar\right)\sin\left(\frac{3\hbar}{2}\right)$.

The third degree polynomial B-spline at the same grid points is as

Consider hybrid cubic B-spline having order three at the same grid information given above is

$$HB_m(\varrho) = gB_m(\varrho) + (1-g)TB_m(\varrho),$$

where value of g plays a significant role that the basis function becomes cubic B-spline if $g = 1$ and it reduces to trigonometric spline as value of g becomes zero. The knots value of $HB_m(\varrho)$ attains value are as follow:

$$TB_m\left(\varrho\right) = \frac{1}{\delta}\begin{cases} q^3\left(\varrho_{m-2}\right); & \text{if} \quad \varrho \in [\varrho_{m-2}, \varrho_{m-1}) \\ p\left(\varrho_{m+2}\right)q^2\left(\varrho_{m-1}\right) + q^2\left(\varrho_{m-2}\right)p\left(\varrho_m\right) + q\left(\varrho_{m-2}\right)q\left(\varrho_{m-1}\right)p\left(\varrho_{m+1}\right); & \text{if} \quad \varrho \in [\varrho_{m-1}, \varrho_m) \\ q\left(\varrho_{m-2}\right)p^2\left(\varrho_{m+1}\right) + p^2\left(\varrho_{m+2}\right)q\left(\varrho_m\right) + q\left(\varrho_{m-1}\right)p\left(\varrho_{m+1}\right)p\left(\varrho_{m+2}\right); & \text{if} \quad \varrho \in [\varrho_m, \varrho_{m+1}) \\ p^3\left(\varrho_{m+2}\right); & \text{if} \quad \varrho \in [\varrho_{m+1}, \varrho_{m+2}) \\ 0; & \text{otherwise.} \end{cases} \tag{2.7}$$

$$B_m\left(\varrho\right) = \frac{1}{6\hbar^3}\begin{cases} (\varrho - \varrho_m)^3; & \text{if} \quad \varrho \in [\varrho_m, \varrho_{m+1}) \\ \hbar^3 + 3\hbar^2\left(\varrho - \varrho_{m+1}\right) + 3\hbar(\varrho - \varrho_{m+1})^2 - 3(\varrho - \varrho_{m+1})^3; & \text{if} \quad \varrho \in [\varrho_{m+1}, \varrho_{m+2}) \\ \hbar^3 + 3\hbar^2\left(\varrho_{m+3} - \varrho\right) + 3\hbar(\varrho_{m+3} - \varrho)^2 - 3(\varrho_{m+3} - \varrho)^3; & \text{if} \quad \varrho \in [\varrho_{m+2}, \varrho_{m+3}) \\ (\varrho_{m+4} - \varrho)^3; & \text{if} \quad \varrho \in [\varrho_{m+3}, \varrho_{m+4}) \\ 0; & \text{otherwise.} \end{cases} \tag{2.8}$$

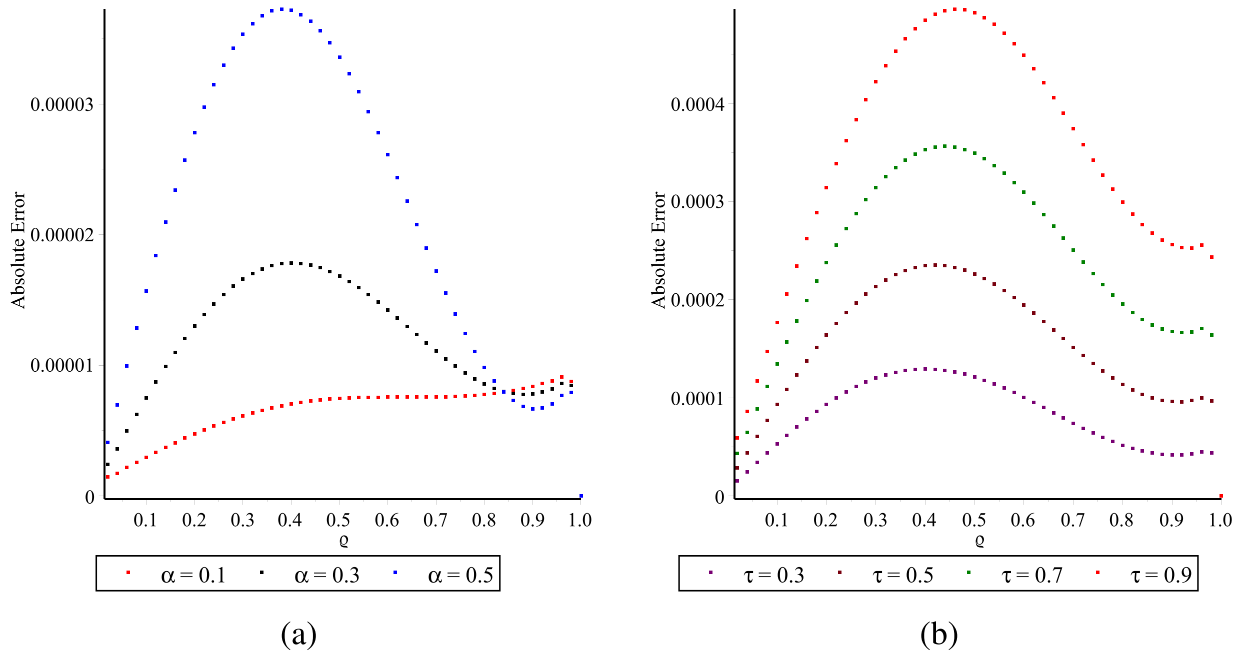

Fig. 1. Geometrical representation of absolute error of Problem 1 for $N = 50$. (a) For $\tau = 0.1$ (b) For $\alpha = 0.5$.

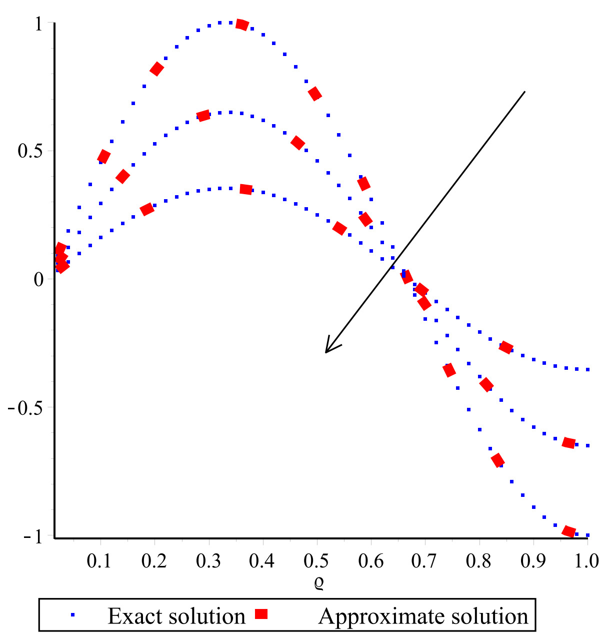

Fig. 2. Comparison of exact solution with approximate solution of Problem 1 for $\tau = 1, 0.7, 0.5$, at $\alpha = 0.5$ and $N = 50$.

Table 5
Absolute errors of Problem 2 at $\alpha = 1, M = 50, N = 50$, and $\tau = 0.01$.

ϱ	$g = 0$	$g = 0.5$	$g = 1$
0.10	1.95030×10^{-7}	2.15912×10^{-2}	0.139834
0.30	5.85084×10^{-7}	2.02839×10^{-2}	0.074794
0.50	9.75010×10^{-7}	1.70270×10^{-3}	0.000741
0.70	1.45382×10^{-6}	2.42978×10^{-2}	0.740884
0.90	1.22623×10^{-6}	1.77999×10^{-2}	0.140172

Table 6
Numerical display of absolute error of Problem 2 at $\alpha = 1$ when $M = 50, N = 50$

τ	ϱ	Exact	MHB-DQM	Error
0.5	0.10	0.066666	0.066723	$5.725145e^{-5}$
	0.30	0.133333	0.133444	$1.111689e^{-4}$
	0.50	0.333333	0.333552	$2.189718e^{-4}$
	0.70	0.466666	0.466877	$2.104338e^{-4}$
	0.90	0.600000	0.600102	$1.020083e^{-4}$
0.7	0.10	0.058823	0.058878	$5.511327e^{-5}$
	0.30	0.176470	0.176622	$1.522951e^{-4}$
	0.50	0.294117	0.294327	$2.100689e^{-4}$
	0.70	0.411764	0.411965	$2.012282e^{-4}$
	0.90	0.529411	0.529508	$9.716124e^{-5}$
1	0.10	0.090909	0.090923	$1.447746e^{-5}$
	0.30	0.272727	0.272768	$4.161873e^{-5}$
	0.50	0.454545	0.454607	$6.184568e^{-5}$
	0.70	0.636363	0.636429	$6.552550e^{-5}$
	0.90	0.818181	0.818217	$3.552343e^{-5}$

Table 7
Approximate solutions for various values of α for Problem 2 when $N = 100$, $M = 60$, $d\tau = 0.01$ and $\tau = 1$.

ϱ	$\alpha = 0.25$	$\alpha = 0.5$	$\alpha = 0.75$
0.1	0.0532717401	0.0527075407	0.0516853593
0.2	0.1063769930	0.1052784431	0.1032844216
0.3	0.1591432605	0.1575708717	0.1547078411
0.4	0.2113856339	0.2094322700	0.2058600013
0.5	0.2628999893	0.2606934849	0.2566356405
0.6	0.3134553615	0.3111621931	0.3069161443
0.7	0.3627852250	0.3606153934	0.3565654040
0.8	0.4105773494	0.4087906853	0.4054251169
0.9	0.4564618718	0.4553760279	0.4533094007

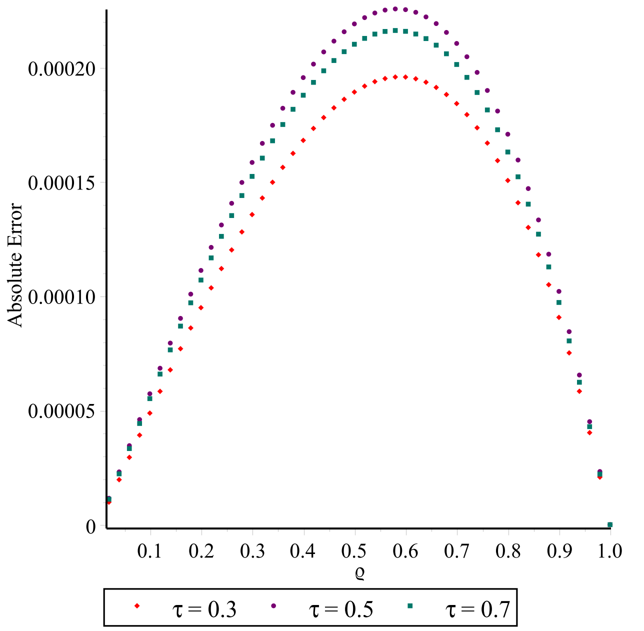

Fig. 3. Geometrical representation of absolute errors of Problem 2 for different values of $\tau, \alpha = 1$ and $N = 50$.

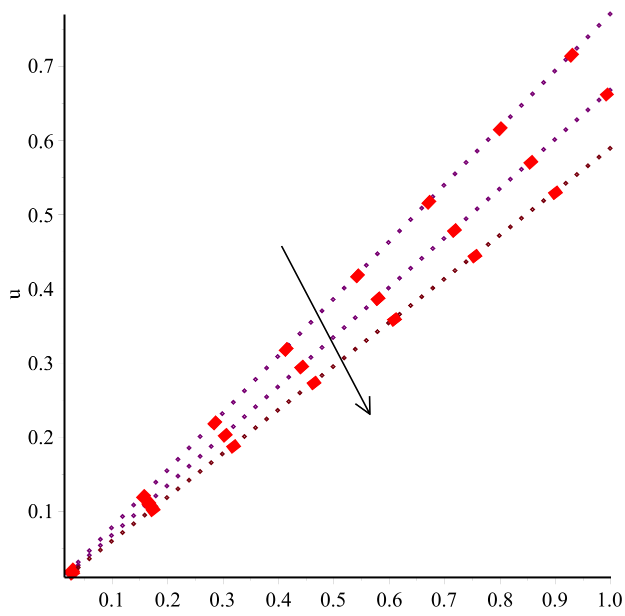

Fig. 4. Comparision of exact solution with approximate solution of Problem 2 for $\tau = 0.3, 0.5, 0.7, \alpha = 1$ and $N = 50$.

Table 8
Absolute errors of Problem 3 for $M = 50, N = 50, d\tau = 0.01$, and $\alpha = 0.5$.

ϱ	$g = 0$	$g = 0.5$	$g = 1$
0.10	$9.2501e^{-6}$	$7.05253e^{-5}$	$4.15017e^{-5}$
0.30	$1.08596e^{-5}$	$7.58392e^{-5}$	$4.72778e^{-5}$
0.50	$4.8693e^{-63}$	$2.18829e^{-61}$	$1.23658e^{-60}$
0.70	$1.08596e^{-5}$	$7.58392e^{-5}$	$4.72778e^{-5}$
0.90	$9.2501e^{-6}$	$8.28048e^{-5}$	$4.15017e^{-5}$

Table 9
Numerical display of error of Problem 3 at $\alpha = 0.5$ when $N = 100$ and $\tau = 1$.

M	ϱ	Exact	MHB-DQM	Error
30	0.10	0.951056	0.951770	$7.137291e^{-4}$
	0.30	0.587785	0.5886692	$8.845346e^{-4}$
	0.50	$2.296e^{-60}$	$2.007e^{-58}$	$1.977738e^{-58}$
	0.70	-0.5877	-0.588631	$8.845346e^{-4}$
	0.90	-0.9510	-0.951726	$7.137291e^{-4}$
50	0.10	0.951056	0.9519102	$9.495576e^{-4}$
	0.30	0.587785	0.5887456	$9.601300e^{-4}$
	0.50	$2.296e^{-60}$	$2.1264e^{-60}$	$1.701539e^{-61}$
	0.70	-0.5877	-0.588765	$9.601300e^{-4}$
	0.90	-0.9510	-9.519125	$8.537666e^{-4}$
70	0.10	0.951056	0.9519372	$8.807802e^{-4}$
	0.30	0.587785	0.5887599	$9.746769e^{-4}$
	0.50	$2.296e^{-60}$	$-1.688e^{-57}$	$1.690656e^{-57}$
	0.70	-0.5877	-0.588732	$9.746769e^{-4}$
	0.90	-0.9510	-0.951979	$7.984682e^{-4}$

Table 10
Numerical display of error of Problem 3 at $N = 50$ when $\tau = 1$.

α	ϱ	Exact	MHB-DQM	Error
0.25	0.10	0.951056	0.9812176	$3.016115e^{-2}$
	0.30	0.587785	0.6464059	$5.862068e^{-2}$
	0.50	$2.296e^{-60}$	$-3.075e^{-59}$	$3.304634e^{-59}$
	0.70	-0.5877	-0.664067	$5.862068e^{-2}$
	0.90	-0.9510	-9.812136	$3.016115e^{-2}$
0.75	0.10	0.951056	0.9319893	$1.906739e^{-2}$
	0.30	0.587785	0.5418891	$4.589585e^{-2}$
	0.50	$2.296e^{-60}$	$6.120e^{-59}$	$5.890384e^{-59}$
	0.70	-0.5877	-0.541856	$4.589585e^{-2}$
	0.90	-0.9510	-0.931992	$1.906739e^{-2}$
0.9	0.10	0.951056	0.9146401	$3.641245e^{-2}$
	0.30	0.587785	0.5061482	$8.163693e^{-2}$
	0.50	$2.296e^{-60}$	$-1.950e^{-59}$	$2.179615e^{-59}$
	0.70	-0.5877	-0.506129	$8.163693e^{-2}$
	0.90	-0.9510	-0.914626	$3.641245e^{-2}$

Table 11
Numerical display of error of Problem 3 at $M = 70$ when $\alpha = 0.5$

$d\tau$	ϱ	Exact	MHB-DQM	Error
0.1	0.10	0.951056	0.9496091	$1.4472901e^{-3}$
	0.30	0.587785	0.5837325	$4.0530463e^{-3}$
	0.50	$2.296e^{-60}$	$-1.492e^{-57}$	$1.495096e^{-57}$
	0.70	-0.5877	-5.837212	$4.053046e^{-3}$
	0.90	-0.9510	-0.934475	$1.789945e^{-3}$
0.05	0.10	0.951056	0.9508718	$1.846557e^{-4}$
	0.30	0.587785	0.5864466	$1.338366e^{-3}$
	0.50	$2.296e^{-60}$	$-3.230e^{-58}$	$3.253299e^{-58}$
	0.70	-0.5877	-0.586432	$1.338366e^{-3}$
	0.90	-0.9510	-0.950871	$1.846557e^{-4}$
0.025	0.10	0.951056	0.9515314	$4.754779e^{-4}$
	0.30	0.587785	0.5878772	$9.224810e^{-5}$
	0.50	$2.296e^{-60}$	$-1.584e^{-57}$	$1.586804e^{-57}$
	0.70	-0.58778	-0.587751	$9.224810e^{-5}$
	0.90	-0.9510	-0.951589	$4.754779e^{-4}$

Table 13
Numerical display of error of Problem 3 at $\alpha = 0.5$ when $M = 50, N = 50$ and $\tau = 0.1$.

μ	ϱ	Exact	MHB-DQM	Error
0.1	0.10	$9.51056e^{-3}$	$1.05773e^{-2}$	$1.06681e^{-3}$
	0.30	$5.87785e^{-3}$	$7.01363e^{-3}$	$1.13578e^{-3}$
	0.50	$2.29615e^{-62}$	$3.43949e^{-61}$	$3.20987e^{-61}$
	0.70	$-5.8775e^{-3}$	$-7.0136e^{-3}$	$1.13578e^{-3}$
	0.90	$-9.5105e^{-3}$	$-1.0577e^{-2}$	$1.06681e^{-3}$
0.5	0.1	$9.51055e^{-3}$	$9.84286e^{-3}$	$3.32295e^{-4}$
	0.30	$5.87785e^{-3}$	$6.27768e^{-3}$	$3.99831e^{-4}$
	0.50	$2.2961e^{-62}$	$-3.7456e^{-61}$	$3.97526e^{-61}$
	0.70	$-5.8778e^{-3}$	$-6.2776e^{-3}$	$4.21317e^{-4}$
	0.90	$-9.5105e^{-3}$	$-9.8428e^{-3}$	$3.32295e^{-4}$
1	0.1	$9.51056e^{-3}$	$9.68893e^{-3}$	$1.78367e^{-4}$
	0.30	$5.87785e^{-3}$	$6.09657e^{-3}$	$2.18721e^{-4}$
	0.50	$-2.29615e^{-62}$	$-2.2269e^{-60}$	$2.24986e^{-60}$
	0.70	$-5.8778e^{-3}$	$-6.0965e^{-3}$	$2.18726e^{-4}$
	0.90	$-9.5105e^{-3}$	$-9.6889e^{-3}$	$1.78367e^{-4}$
0.0001	0.1	$9.51055e^{-1}$	1.072684	$1.21625e^{-1}$
	0.30	$5.87785e^{-1}$	$0.68246e^{-3}$	$9.946828e^{-4}$
	0.50	$2.2961e^{-60}$	$3.2000e^{-58}$	$3.17703e^{-58}$
	0.70	$-5.8778e^{-1}$	$-6.82468e^{-1}$	$9.46828e^{-2}$
	0.90	$-9.5105e^{-1}$	$-9.12792e^{-1}$	$1.03775e^{-1}$

Table 12
Comparison of approximated solution of Problem 3 at $M = 40$ when $\alpha = 0.5$ and $\tau = 1$.

ϱ	Exact	MHB-DQM	Esen[16]
0.10	0.951056	0.951870	0.951005
0.20	0.809016	0.8101060	0.808954
0.30	0.587785	0.5887239	0.587738
0.40	0.309016	0.309545	0.308993
0.50	0.000000	0.000000	0.000000
0.60	−0.309016	−0.309545	−0.308996
0.70	−0.587785	−0.588723	−0.587741
0.80	−0.809016	−0.810106	−0.808957
0.90	−0.951056	−0.951870	−0.951008

$$HB_m\left(\varrho\right) = \begin{cases} \dfrac{g}{6} + \left(1-g\right)csc\left(\hbar\right)sin^2\left(\dfrac{\hbar}{2}\right)csc\left(\dfrac{3\hbar}{2}\right); & \\ \dfrac{2g}{3} + \left(1-g\right)\dfrac{2}{cos(\hbar)+1}; & i=m \\ 0; & \text{otherwise.} \end{cases} \tag{2.9}$$

and the derivative of $HB_m(\varrho)$ at each knots is as following

$$HB'_m\left(\varrho_i\right) = \begin{cases} \dfrac{g}{2\hbar} + \left(1-g\right)\dfrac{3}{4}csc\left(\dfrac{3\hbar}{2}\right); & i=m-1 \\ \dfrac{-g}{2\hbar} - \left(1-g\right)\dfrac{3}{4}\dfrac{2}{cos(\hbar)+1}; & i=m+1 \\ 0; & \text{otherwise.} \end{cases} \tag{2.10}$$

Since the nodal points $\varrho_{-1}, \varrho_{M+1}$ lie outsides the space domain so a weighted relation is used to evaluate the B-spline outside its support. This procedure was first incorporated by Mital and Jian [34] for cubic B-spline and Zhu et al. [35] for cubic trigonometric spline.

Hybrid cubic spline is modified as

$$\begin{aligned} &MHB_0(\varrho) = HB_0(\varrho) + 2HB_{-1}(\varrho),\\ &MHB_1(\varrho) = HB_1(\varrho) - HB_{-1}(\varrho),\\ &MHB_m(\varrho) = HB_m(\varrho); \qquad m = 2,\dots,M-2,\\ &MHB_{M-1}(\varrho) = HB_{M-1}(\varrho) - HB_{M+1}(\varrho),\\ &MHB_M(\varrho) = HB_M(\varrho) + 2HB_{M+1}(\varrho). \end{aligned}$$

Description of proposed method

Bellman et al. had developed the DQM, to calculate the spatial derivative of function by utilizing a weighted sum of functional values at certain grid points. In DQM, we can use various types of test functions to calculate the weighted coefficients, like Spline function, Legendre polynomial, Lagrange interpolation polynomials etc but here we have used modified hybrid cubic B-Spline as a test function. We consider a partition $a \leqslant \varrho \leqslant b$, where $[a, b]$ is that the problem domain as $a = \varrho_1 < \varrho_2 < \cdots < \varrho_N = b$ be the knots and $\hbar = \varrho_{i+1} - \varrho_i, i = 1, 2, \dots, M-1, M$.

According to DQM, partial derivative of first and second order of function $u(\varrho, \tau)$ at any time τ can be calculated as

$$\frac{\partial^r u(\varrho_i, \tau)}{\partial \varrho^r} \cong \sum_{j=0}^{M} a_{ij}^{(r)} u\left(\varrho_j, \tau\right), \quad 0 \leqslant i \leqslant M. \tag{3.1}$$

where $r \in Z^+, 0 \leqslant i, j \leqslant M$ and $a_{ij}^{(r)}$ $(r = 1, 2)$ are allowing to calculate the derivatives at ϱ of $u(\varrho, \tau)$ in DQM at specified grid points. To determine the weighted coefficients, we have used modified cubic B-spline basis function.

Calculation of weighted coefficients

To calculate unknown weights in one dimension, we apply $MHB_m(\varrho)_{m=0}^{M}$, for this put $r = 1$ and using the B-spline function (3.1), we got

$$\frac{\partial MHB_m(\varrho_i)}{\partial \varrho} = \sum_{j=0}^{M} a_{ij}^{(1)} MHB_m\left(\varrho_j\right); \qquad 0 \leqslant i, \quad m \leqslant M.$$

First order derivative of weighted coefficients $a_{ij}^{(1)}$ is yet to calculate. By using properties above equations give in matrix form

$$\begin{cases} Pa_0^{(1)} = A_0, \\ Pa_1^{(1)} = A_1, \\ . \\ . \\ . \\ Pa_K^{(1)} = A_K. \end{cases} \tag{4.1}$$

where, P is the coefficient matrix of order $(M+1) \times (M+1)$.

$$P = \begin{pmatrix} P_0 + 2P_1 & P_1 & 0 & . & .. & \dots & \dots & 0 \\ 0 & P_0 & P_1 & 0 & ..\ddots & & & \\ 0 & P_1 & P_0 & P_1 & \ddots & & & \\ \vdots & & & & \ddots & & & \vdots \\ \vdots & & & & & P_1 & P_0 & 0 \\ 0 & \dots & \dots & \dots & \dots & & P_1 & P_0 + 2P_1 \end{pmatrix}, \tag{4.2}$$

where

$P_0 = \frac{2g}{3} + \left(1-g\right)\frac{2}{1+2cos(\hbar)}$, $\quad P_1 = \frac{g}{6} + \left(1-g\right)sin^2\left(\frac{\hbar}{2}\right)csc\left(\frac{3\hbar}{2}\right)csc(\hbar)$, at ϱ_k and $a_k^{(1)}$, $0 \leqslant k \leqslant M$ are weighted coefficients vectors is to be determined i.e.

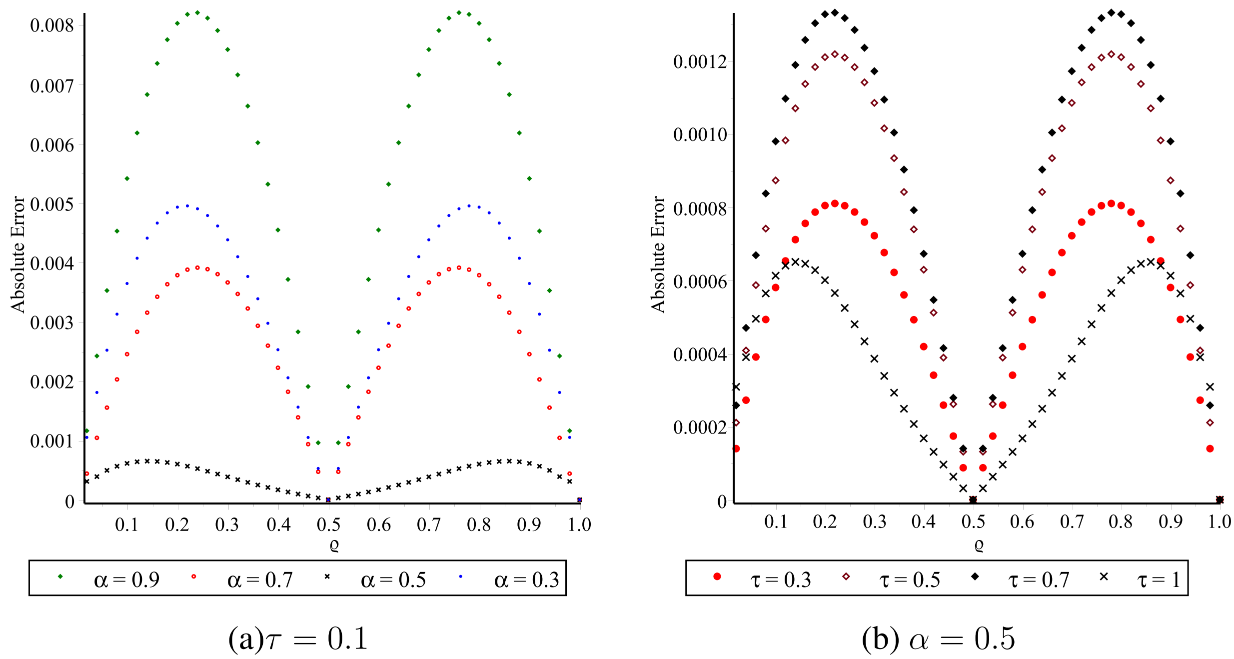

Fig. 5. Geometrical representation of absolute errors of Problem 3 at $N = 50$.

$$a_k^{(1)} = \left[a_{k0}^{(1)}, a_{k1}^{(1)}, a_{k2}^{(1)}, \ldots, a_{kM}^{(1)}\right]^T,$$

the right side vector A_k at ϱ_k, $0\leqslant k\leqslant M$ is as follows

$$\begin{aligned} A_0 &= [-2z, 2z, 0, \ldots, 0, 0, 0]^T, \\ A_1 &= [-z, 0, z, 0, \ldots, 0, 0, 0]^T, \\ &\vdots \\ A_{M-1} &= [0, 0, 0, \ldots, -z, 0, z]^T, \\ A_M &= [0, 0, 0, \ldots, 0, -2z, 2z]^T. \end{aligned}$$

with $z = \frac{g}{2\hbar} + \left(\frac{3}{4}\right)\left(1-g\right)csc\left(\frac{3\hbar}{2}\right)$.

The value of $a_{ij}^{(1)}$ can be calculated by using (4.1). The weighted coefficients of $a_{ij}^{(2)}$ can be calculated by using recursive formula given by Richard and Shu [36], $r\geqslant 2$ i.e.

$$a_{ij}^{(r)} = m\left(a_{ii}^{(r-1)}a_{ij}^{(1)} - \frac{a_{ij}^{(r-1)}}{\varrho_i - \varrho_j}\right), \quad i \neq j, \quad 0\leqslant i\leqslant M,$$

$$a_{ij}^{(r)} = -\sum_{j=0, j\neq i}^{M} a_{ij}^{(r)}, \quad i = j.$$

where, $a_{ij}^{(r-1)}$ and $a_{ij}^{(r)}$ are weighted coefficients of $(r-1)^{th}$ and r^{th} order derivatives with respect to ϱ.

Description of algorithm

In this subsection, a DQ method which is based on modified hybrid B-Spline (MHB) is constructed for the solution of Eq. (1.1).

In order to design the algorithm, DQM is used as on (1.1)–(1.2)

$$D_\tau^\alpha u(\varrho,\tau) + u(\varrho,\tau)D_\varrho u(\varrho,\tau) - \mu D_\varrho^2 u(\varrho,\tau) = 0.$$

Substituting the weighted sum (3.1) in above equation, gives the following form:

$$D_\tau^\alpha u\left(\varrho,\tau\right) + u\left(\varrho,\tau\right)\left[\sum_{j=0}^{M} a_{ij}^{(1)} u\left(\varrho_j,\tau\right)\right] - \mu\left[\sum_{j=0}^{M} a_{ij}^{(2)} u\left(\varrho_j,\tau\right)\right] = 0, \tag{4.3}$$

with $i = 0, 1, 2, \ldots M$.

By discretizing the above equation in the presence of Caputo formula at $u(\varrho,\tau_n)$, we have

$$\begin{aligned} &\frac{1}{d\tau^\alpha}\sum_{k=0}^{n-1} w_k^\alpha u\left(\varrho_i,\tau_{n-k}\right) - \frac{1}{d\tau^\alpha}\sum_{k=0}^{n-1} w_k^\alpha u\left(\varrho_i,0\right) + u\left(\varrho_i,\tau_n\right)\sum_{j=1}^{M-1} a_{ij}^{(1)} u\left(\varrho_j,\tau_n\right) - \mu\sum_{j=1}^{M-1} a_{ij}^{(2)} u\left(\varrho_j,\tau_n\right) + u\left(\varrho_i,\tau_n\right)\left[a_{i0}^{(1)}\phi_1(\tau_n) + a_{iM}^{(1)}\phi_2(\tau_n)\right] - \mu\left[a_{i0}^{(2)}\phi_1(\tau_n)\right. \\ &\left. + a_{iM}^{(2)}\phi_2(\tau_n)\right] \\ &= 0. \end{aligned} \tag{4.4}$$

Above equation can also be written as

$$\frac{1}{d\tau^{\alpha}}\sum_{k=0}^{n-1} w_k^{\alpha} u\left(\varrho_i,\tau_{n-k}\right) - \frac{1}{d\tau^{\alpha}}\sum_{k=0}^{n-1} w_k^{\alpha} u\left(\varrho_i,0\right) + u\left(\varrho,\tau_n\right)\sum_{j=1}^{M-1} a_{ij}^{(1)} u\left(\varrho_j,\tau_n\right) - \mu\sum_{j=1}^{M-1} a_{ij}^{(2)} u\left(\varrho_j,\tau_n\right) + G_i^n = 0, \tag{4.5}$$

where

$$G_i^n = u(\varrho,\tau_n)\left[a_{i0}^{(1)}\phi_1(\tau) + a_{iM}^{(1)}\phi_2(\tau)\right] - \mu\left[a_{i0}^{(2)}\phi_1(\tau) + a_{iM}^{(2)}\phi_2(\tau)\right].$$

By simplifying and rearranging the above equation, we get

$$w_0^{\alpha} u_i^n + u_i^n d\tau^{\alpha}\sum_{j=1}^{M-1} a_{ij}^{(1)} u_j^n - d\tau^{\alpha}\mu\sum_{j=1}^{M-1} a_{ij}^{(2)} u_j^n = \sum_{k=0}^{n-1} w_k^{\alpha} u_i^0 - d\tau^{\alpha} G_i^n - \sum_{k=1}^{n-1} w_k^{\alpha} u_i^{n-k}, \tag{4.6}$$

While varrying $i = 1, 2, \ldots M-1$, we have

$$w_0^{\alpha}\begin{pmatrix} u_1^n \\ u_2^n \\ \vdots \\ u_{M-1}^n \end{pmatrix} + d\tau^{\alpha}\begin{pmatrix} u_1^n \\ u_2^n \\ \vdots \\ u_{M-1}^n \end{pmatrix}\begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1,M-1}^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & \cdots & a_{2,M-1}^{(1)} \\ \vdots & \vdots & \cdots & \vdots \\ a_{M-1,1}^{(1)} & a_{M-1,2}^{(1)} & \cdots & a_{M-1,M-1}^{(1)} \end{pmatrix}\begin{pmatrix} u_1^n \\ u_2^n \\ \vdots \\ u_{M-1}^n \end{pmatrix}$$

$$-\mu\begin{pmatrix} a_{11}^{(2)} & a_{12}^{(2)} & \cdots & a_{1,M-1}^{(2)} \\ a_{21}^{(2)} & a_{22}^{(2)} & \cdots & a_{2,M-1}^{(2)} \\ \vdots & \vdots & \cdots & \vdots \\ a_{M-1,1}^{(2)} & a_{M-1,2}^{(2)} & \cdots & a_{M-1,M-1}^{(2)} \end{pmatrix}\begin{pmatrix} u_1^n \\ u_2^n \\ \vdots \\ u_{M-1}^n \end{pmatrix}$$

$$= \sum_{k=0}^{n-1} w_k^{\alpha}\begin{pmatrix} u_1^0 \\ u_2^0 \\ \vdots \\ u_{M-1}^0 \end{pmatrix} - d\tau^{\alpha}\begin{pmatrix} G_1^n \\ G_2^n \\ \vdots \\ G_{M-1}^n \end{pmatrix} - \sum_{k=1}^{n-1} w_k^{\alpha}\begin{pmatrix} u_1^{n-k} \\ u_2^{n-k} \\ \vdots \\ u_{M-1}^{n-k} \end{pmatrix}.$$

In compact form,

$$w_0^{\alpha} U^n + d\tau^{\alpha} U^n W^{(1)} U^n - d\tau^{\alpha}\mu W^{(2)} U^n = \sum_{k=0}^{n-1} w_k^{\alpha} U^0 - d\tau^{\alpha} G^n - \sum_{k=1}^{n-1} w_k^{\alpha} U^{n-k}, \tag{4.7}$$

where

$$W^{(1)} = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1,M-1}^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & \cdots & a_{2,M-1}^{(1)} \\ \vdots & \vdots & \cdots & \vdots \\ a_{M-1,1}^{(1)} & a_{M-1,2}^{(1)} & \cdots & a_{M-1,M-1}^{(1)} \end{pmatrix},$$

$$W^{(2)} = \begin{pmatrix} a_{11}^{(2)} & a_{12}^{(2)} & \cdots & a_{1,M-1}^{(2)} \\ a_{21}^{(2)} & a_{22}^{(2)} & \cdots & a_{2,M-1}^{(2)} \\ \vdots & \vdots & \cdots & \vdots \\ a_{M-1,1}^{(2)} & a_{M-1,2}^{(2)} & \cdots & a_{M-1,M-1}^{(2)} \end{pmatrix}.$$

For initial state, IC given in (1.1) is used. After this, the approximation of unknown function under consideration is obtained by performing iteration process.

Stability analysis

In this section, we have explained the numerical stability of proposed scheme.

Lemma 1. *Let U^n be the approximation of the exact solution $\widetilde{U}^n$, then*

$$e^n = \sum_{k=0}^{n-1} w_k^{\alpha} e^0 (I + d\tau^{\alpha} K)^{-1} - \sum_{k=1}^{n-1} w_k^{\alpha} e^{n-k}(I + d\tau^{\alpha} K)^{-1}.$$

Proof. Let U^n be defined as $U^n = U(\varrho_i, \tau_n)$ then the hybrid-B-spline DQM leads to the difference Eq. (1.1)

$$w_0^{\alpha} U^n + U^n d\tau^{\alpha} W^{(1)} U^n - d\tau^{\alpha}\mu W^{(2)} U^n = \sum_{k=0}^{n-1} w_k^{\alpha} U^0 - d\tau^{\alpha} G^n - \sum_{k=1}^{n-1} w_k^{\alpha} U^{n-k}.$$

By simplifying the above equation, we get

$$U^n\left[w_0^{\alpha} + d\tau^{\alpha} W^{(1)} U^n - d\tau^{\alpha}\mu W^{(2)}\right] = \sum_{k=0}^{n-1} w_k^{\alpha} U^0 - d\tau^{\alpha} G^n - \sum_{k=1}^{n-1} w_k^{\alpha} U^{n-k}. \tag{5.1}$$

Let

$$U^n W^{(1)} - \mu W^{(2)} = K,$$

Then the above equation becomes

$$U^n\left[w_0^{\alpha} + d\tau^{\alpha} K\right] = \sum_{k=0}^{n-1} w_k^{\alpha} U^0 - d\tau^{\alpha} G^n - \sum_{k=1}^{n-1} w_k^{\alpha} U^{n-k}. \tag{5.2}$$

Let U^0 be the approximation to the initial vector $\widetilde{U}^0$, then we have

$$\widetilde{U}^n\left[w_0^{\alpha} + d\tau^{\alpha} K\right] = \sum_{k=0}^{n-1} w_k^{\alpha} \widetilde{U}^0 - d\tau^{\alpha} \widetilde{G}^n - \sum_{k=1}^{n-1} w_k^{\alpha} \widetilde{U}^{n-k}. \tag{5.3}$$

Subtracting (5.3) from (5.2), and taking $e^n = U^n - \widetilde{U}^n$ we get,

$$e^n\left[I + d\tau^{\alpha} K\right] = \sum_{k=0}^{n-1} w_k^{\alpha} e^0 - \sum_{k=1}^{n-1} w_k^{\alpha} e^{n-k}.$$

$\Rightarrow$

$$e^n = \sum_{k=0}^{n-1} w_k^{\alpha} e^0 (I + d\tau^{\alpha} K)^{-1} - \sum_{k=1}^{n-1} w_k^{\alpha} e^{n-k}(I + d\tau^{\alpha} K)^{-1}. \tag{5.4}$$

where, I is identity matrix in same dimension as of K matrix.

Theorem 1. *Let $e^n, n = 0, 1, 2....m$ be the solution of (5.4), we have* $\|e^n\| \leqslant \|e^0\|$.

Proof. In order to prove, $\|e^n\| \leqslant \|e^0\|$, we have preferred mathematical induction process.

Let for $n = 1$, (5.4) will have the form

$$e^1 = \sum_{k=0}^{1-1} w_k^\alpha e^0 (I + d\tau^\alpha K)^{-1} - \sum_{k=1}^{1-1} w_k^\alpha e^{1-k} (I + d\tau^\alpha K)^{-1}.$$

Taking norm on both sides

$$\|e^1\| = \|w_0^\alpha e^0 (I + d\tau^\alpha K)^{-1}\|$$

$$\leqslant \|w_0^\alpha (I + d\tau^\alpha K)^{-1}\| \|e^0\|. \tag{4.5}$$

Assume that $\left\|(I + d\tau^\alpha K)^{-1}\right\| \leqslant 1$, and $\|w_0^\alpha\| = 1$so above equation will have the form

$$\|e^1\| \leqslant \|e^0\|,$$

Suppose that (5.4) is true for $m = n-1$, i.e.

$$\|e^m\| \leqslant \|e^0\|, m = 1, 2, ...n-1 \tag{4.6}$$

Now consider

$$\|e^n\| = \|\sum_{k=0}^{n-1} w_k^\alpha e^0 (I + d\tau^\alpha K)^{-1} - \sum_{k=1}^{n-1} w_k^\alpha e^{n-k} (I + d\tau^\alpha K)^{-1}\|.$$

Using (4.6) the above have the form

$$\|e^n\| \leqslant \|(I + d\tau^\alpha K)^{-1}\| \cdot \| \sum_{k=0}^{n-1} w_k^\alpha - \sum_{k=1}^{n-1} w_k^\alpha \| \max_{0 \leqslant m \leqslant n-1} \|e^m\|.$$

or

$$\|e^n\| \leqslant \|(I + d\tau^\alpha K)^{-1}\| \cdot \|e^0\|.$$

By using our assumption we have,

$$\|e^n\| \leqslant \|e^0\|.$$

which shows that the proposed method is stable.

If assumption does not hold i.e $\left\|(I + d\tau^\alpha K)^{-1}\right\| \geqslant 1$, then there always exists a suitable choice of the involved parameters, which will help to hold the condition i.e

$$\|(I + d\tau^\alpha K)^{-1}\| \leqslant 1.$$

Numerical experiments

In this section, we have presented the numerical results of MHB-DQM on non-linear time-fractional Burger's equations. Here, we use the following error norms to examine the accuracy of method.

Absolute error $= |(u_{exact})_i - U_i|$.

$$L_2 = \|u_{exact} - U\|_2 \simeq \sqrt{\hbar \sum_{i=0}^{M} \left|(u_{exact})_i - U_i\right|^2},$$

and

$$L_\infty = \|u_{exact} - U\|_\infty \simeq max_{i=0}^{M} |(u_{exact})_i - U_i|.$$

Problem 1

Consider the following time-fractional Burger's equation: ([10])

$$\frac{\partial^\alpha u(\varrho, \tau)}{\partial \tau^\alpha} + u\left(\varrho, \tau\right) \frac{\partial u(\varrho, \tau)}{\partial \varrho} - \mu \frac{\partial^2 u(\varrho, \tau)}{\partial \varrho^2} = H\left(\varrho, \tau\right), \tag{6.1}$$

with following conditions

$$u\left(\varrho, 0\right) = 0, \quad 0 \leqslant \varrho \leqslant 1, \quad u\left(0, \tau\right) = 0, \quad u\left(1, \tau\right) = -\tau^{\frac{3}{2}}, \quad \tau \geqslant 0.$$

where,

$$H\left(\varrho, \tau\right) = \frac{\tau^{\frac{3}{2}-\alpha}}{\Gamma\left(\frac{5}{2} - \alpha\right)} \Gamma\left(\frac{5}{2}\right) \sin\left(\frac{3\pi}{2}\left(\varrho\right)\right) + \frac{3\pi}{2} \tau^3 \sin\left(\frac{3\pi}{2}\left(\varrho\right)\right)\left(\cos\frac{3\pi}{2}\left(\varrho\right)\right) + \frac{9\pi^2}{4}\left(\tau^{\frac{3}{2}}\right) \sin\left(\frac{3\pi}{2}\left(\varrho\right)\right).$$

In order to see, which spline provides the better result, solution have been oobtained for the different values of g. It is evident from Table 1, that trigonometric spline provides the better result. Absolute error for $\alpha = 0.5, 0.9, M = 50$ and $N = 50$ for different values of τ are shown in Table 2 and Table 3 through, which we come to know the that decreasing values of τ provides the better convergent solution. Table 4 presents the comparison of exact and MHB-DQM solution.

Graph of absolute error for $\alpha = 0.1, 0.3, 0.5$ at $\tau = 0.1$ are shown in Fig. 1 (a), which shows that the higher values of α provides the better solution. Graph of absolute error for $\tau = 0.3, 0.5, 0.7, 0.9$ at $\alpha = 0.5$ are shown in Fig. 1 (b). The comparison of exact solution with approximate solution showed that solutions are close to the exact one as shown in Fig. 2.

Problem 2

Consider the time fractional Burger's equation: ([13])

$$\frac{\partial^\alpha u(\varrho, \tau)}{\partial \tau^\alpha} + u\left(\varrho, \tau\right) \frac{\partial u(\varrho, \tau)}{\partial \varrho} - \frac{\partial^2 u(\varrho, \tau)}{\partial \varrho^2} = 0, \quad 0 < \alpha < 1. \tag{6.2}$$

with following conditions

$$u\left(\varrho, o\right) = \varrho, \quad 0 \leqslant \varrho \leqslant 1, \quad u\left(0, \tau\right) = 0, \quad u\left(1, \tau\right) = \frac{1}{1 + \tau}, \quad \tau > 0.$$

In order to see, which spline provides the better result, solution have been oobtained for the different values of g. It is evident from Table 5, that trigonometric spline provides the better result. Absolute error for different values of τ when M, N are fixed for $\alpha = 1$ are shown in Table 6 which shows that increasing the value of τ gives better result than others. Table 7 represents the approximate solutions for various values of α when τ is fixed. Graph of absolute error for different value of τ are shown in Fig. 3. Comparison of absolute and exact solution is shown in

Fig. 4 for different values of τ.

Problem 3

Consider the time fractional Burger's equation: ([16])

$$\frac{\partial^\alpha u(\varrho,\tau)}{\partial\tau^\alpha}+u\left(\varrho,\tau\right)\frac{\partial u(\varrho,\tau)}{\partial\varrho}-\mu\frac{\partial^2 u(\varrho,\tau)}{\partial\varrho^2}=H\left(\varrho,\tau\right), \tag{6.3}$$

with following conditions

$$u\left(\varrho,0\right)=0,\qquad 0\leqslant\varrho\leqslant1.\quad u\left(0,\tau\right)=\tau^2,\qquad u\left(1,\tau\right)=-\tau^2,\quad \tau\geqslant0,$$

where,

$$H\left(\varrho,\tau\right)=\frac{2\tau^{2-\gamma}cos\pi\varrho}{\Gamma(3-\gamma)}-\pi\tau^4cos\left(\pi\varrho\right)sin\left(\pi\varrho\right)+\mu\pi^2\tau^2cos\left(\pi\varrho\right).$$

In Table 8, the absolute error for different values of g is calculated and it shows that $g=0$ gives better result, so we have adopted trigonometric B-spline for further calculations. Table 9 represents the absolute error at $\alpha=0.5$ for different values of M where N and τ are fixed which shows that results are symmetric at $M=30$. Absolute error for different values of α, when N is fixed with $\tau=1$ is shown in Table 10. It indicates that results are symmetric and solution is also symmetric for the different values of μ and also for $d\tau=0.05,0.025$ when $M=70$ which is shown in Table 11 and Table 13. A comparison table with results presented by Esen et al. [16] is presented in Table 12. Graph of absolute error for $\alpha=0.1,0.3,0.5$ at $\tau=1$ as shown in Fig. 5 (a), which show that increasing values of α provides the convergent solution. Graph of absolute error for $\tau=0.3,0.5,0.7,0.9$ for $\alpha=0.5$ as shown in Fig. 5 (b). Comparison of exact solution with approximate solution is shown in Table 8.

Concluding remarks

we have proposed a method namely Differential Quadrature method (DQM) to solve time fractional Burger equation. In this method, hybrid version of B-spline blended function with trigonometric and polynomial spline are used along with DQM as basis function to calculate the weighted coefficients. It is noticed that the trigonometric B-spline provides better solution than the polynomial B-spline, so we have adopted trigonometric spline as represented through Table 1, Table 5 and Table 8. The study of numerical results reveals that as the value of τ decreases the approximate solution becomes more convergent as reflected through Table 2, Table 3 and Table 6. It is seen that, if the step size with respect to τ decrease, not much changes in result have been seen. It might have changes for very large values of M (1000say). The smaller the value of fractional parameter α provides better solution than the larger value as seen from the Fig. 1 (a). Figs. 2 and 4 indicated that the results of Problem 1 and 2 are very much in agreement with the exact solution. The results are also compared with the method presented by Esen et al. [16]. The MHB-DQM has proved it's efficiency for the solution of Burger equation. The numerical scheme can be extended for higher order and higher dimensional problems in future.

Funding

National Natural Science Foundation of China (No. 71601072), Key Scientific Research Project of Higher Education Institutions in Henan Province of China (No. 20B110006) and the Fundamental Research Funds for the Universities of Henan Province (No. NSFRF210314).

CRediT authorship contribution statement

Muhammad Sadiq Hashmi: Conceptualization, Methodology. **Misbah Wajiha:** Writing - original draft, Validation. **Shao-Wen Yao:** Writing - original draft, Writing - review & editing. **Abdul Ghaffar:** Methodology. **Mustafa Inc:** Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work is supported by the Higher Education Commission of Pakistan under the NRPU project Grant No. 9306/Punjab/NRPU/R&D/HEC/2017.

References

[1] Batemann H. Some recent researches on the motion of fluids. Mon Weather Rev 1915;43(4):163–70.
[2] Burgers JM. A mathematical model illustrating the theory of turbulence. Adv Appl Mech 1948;1:171–99.
[3] Whitham GB. Linear and Nonlinear Waves. Pure and Applied Mathematics. John Wiley and Sons; 2011. p. 42.
[4] Bluman GW, Cole JD. The general similarity solution of the heat equation. J Math Mech 1969;18:025–1042.
[5] Ford NJ, Conolly JA. Systems-based decomposition schemes for the approximate solution of multi-term fractional differential equations. J Comput Appl Math 2009;229(2):389–91.
[6] Dehghan M, Yousefi SA, Lofti A. The use of He's variational iteration method for solving the telegraph and fractional telegraph equations. Int J Numer Methods Biomed Eng 2011;27(2):219–31.
[7] Mokhtary P. Reconstruction of exponentially rate of convergence to Legendre collocation solution of a class of fractional integro-differential equations. J Comput Appl Math 2015;279:145–58.
[8] Lin Y, Xu C. Finite difference/spectral approximations for the time fractional diffusion equation. J f Comput Phys 2007;225(2):1533–52.
[9] Bellman R, Kashef BG, Casti J. Differential quadrature: a technique for the rapid solution of nonlinear differential equations. J Comput Phy 1972;10:40–52.
[10] Majeed A, Kamran M, Iqbal MK, Baleanu D. Solving time fractional Burger's and Fisher's equation using cubic B-spline approximation method. Adv Diff Eq 2020;2020:175.
[11] Mittal RC, Dahiya S. Numerical simulation on hyperbolic diffusion equations using modified cubic B-spline differential quadrature methods. Comput Math Appl 2015;70(5):737–49.
[12] Dahiya S, Mittal RC. Numerical Solutions of Differential Equations Using Modified B-spline Differential Quadrature Method. In: Springer Proceedings in Mathematics and Statistics; 2015.
[13] Yaseen M, Abbas M. An efficient computational technique based on cubic trigonometric B-splines for time-fractional Burger's equation. Int J Computer Math 2019;97(3):725–38.
[14] Agarwal OP. A general solution for the fourth order fractional diffusion wave equation. Fractional Calculus Appl Anal 2000;3(1):1–12.
[15] Agarwal OP. A general solution for the fourth order fractional diffusion wave equation defined in bounded domain. Computer Struct 2001;79(16):1497–501.
[16] Esen A, Tasbozan O. Numerical solution of time-fractional Burger's equation. Acta UnivActa Univ Sapientiae, Math 2015;7:167–85.
[17] Shukla HS, Tamsir M, Srivastava VK, Kumar J. Numerical solution of two dimensional coupled viscous Burger equation using modified cubic B-spline differential quadrature method. AIP Adv 2014;4(11):117134.
[18] Kuo CK, Lee SY. A new exact solution of Burger's equation with linearized solution. Math Problems Eng 2015;2015:414808.
[19] Abdel-Salam EA-B, Yousif EA, Arko YAS, Gumma EAE. Solution of moving boundary space-time fractional Burger's equation. J Appl Math 2014;2014:218092.
[20] Hassan S, Al-Smadi M, El-Ajou A, Momani S, Hadid S, Al-Zhuor Z. Numerical approach in the Hilbert space to solve a fuzzy Atangana-Baleanu fractional hybrid system. Choas, Solitons Fractals 2021;143:110506.
[21] Omar AA, Momani S, Al-Smadi M. Numerical computations of coupled fractional resonant Schrodinger equations arising in quantum mechanics under conformable fractional derivative sense. Phys Scr 2020;95(7):075218.
[22] Al-Smadi M, Omar AA. Computational algorithm for solving fredholm time-fractional partial integro-differential equations of dirichlet functions type with error estimates. Appl Math Comput 2019;342:280–94.
[23] Hadid S, Omar AA, Al-Smadi M. Approximate solutions of nonlinear fractional Kundu-Eckhaus and coupled fractional massive Thirring equations emerging in quantum field theory using conformable residual power series method. Phys Scr 2020;95(10):105205.
[24] Jiwari R, Kumar S, Mittal RC. Meshfree algorithms based on radial basis functions for numerical simulation and to capture shocks behavior of Burgers' type problems. Eng Comput 2019;36(4):1142–68.

[25] Jiwari R, Srivastav V, Tamsir M. An algorithm based on exponential modified cubic B-spline differential quadrature method for nonlinear Burger equation. Appl Math Comput 2016;290:111–24.

[26] Jiwari R. A hybrid numerical scheme for the numerical solution of the Burgers' equation. Comput Phys Commun 2015;188:59–67.

[27] Jiwari R, Mittal RC, Sharma KK. A numerical scheme based on weighted average differential quadrature method for the numerical solution of Burgers' equation. Appl Math Comput 2014;219:6680–91.

[28] Jiwari R, Mittal RC. A differential quadrature method for numerical solutions of Burgers'-type equations. Int J Numer Meth Heat Fluid Flow 2012;22(7):880–95.

[29] Hashmi MS, Awais M, Waheed A, Ali Q. Numerical treatment of Hunter Saxton equation using cubic trigonometric B-spline collocation method. AIP Adv 2017;7:095124.

[30] Shikrani R, Hashmi MS, Khan N, Ghaffar A, Nisar KS, Singh J, Kumar D. An efficient numerical approach for space fractional partial differential equations. Alexandria Eng J 2020;59(5):2911–9.

[31] Podlubny I. Fractional Differential equations. San Diego, CA: Academic Press; 1999.

[32] Kilbas AA, Srivastava HM, Trujillo JJ. Theory and Applications of Fractional Differential Equations. Amsterdam: Elsevier B.V; 2006.

[33] Chen MH, Deng WH. Fourth order difference approximations for space Reimann-Liouville derivatives based on weighted and shifted Lubich difference operators. Commun Comput Phys 2014;16(2):516–40.

[34] Mittal RC, Jain RK. Numerical solution of non-linear Burger's equation with modified cubic B-Spline collocation method. Appl Math Comput 2012;218(15):7839–55.

[35] Zhu XG, Nie YF, Zhang WW. An efficient differential quadrature method for fractional advection-diffusion equation. Nonlinear Dyn 2017;90(3):1807–27.

[36] Shu C, Richards BE. Application of generalized differential quadrature to solve two-dimensional incompressible Navier-Stokes equations. Int J Numer Methods Fluids 1992;15(7):791–8.