Proc. Natl. Acad. Sci., India, Sect. A Phys. Sci. (July–September 2021) 91(3):461–474
https://doi.org/10.1007/s40010-020-00684-y

RESEARCH ARTICLE

A Comparative Study of Cubic B-spline-Based Quasi-interpolation and Differential Quadrature Methods for Solving Fourth-Order Parabolic PDEs

R. C. Mittal[1] · Sudhir Kumar[1] · Ram Jiwari[1]

Received: 25 September 2018 / Revised: 2 April 2020 / Accepted: 9 May 2020 / Published online: 4 June 2020
© The National Academy of Sciences, India 2020

Abstract In this work, we present two approaches for simulation of fourth-order parabolic partial differential equations. In the first method, cubic B-spline quasi-interpolation is used to approximate the spatial derivative of the dependent variable and forward difference to approximate the time derivative. In the second method, we have used modified cubic B-spline functions-based differential quadrature method (DQM) for space discretization to get a system of ODEs and then this system is solved by SSP-RK43 method to get the results at knots. The numerical results demonstrate the accuracy of the proposed method. The stability analysis of the methods has also been discussed. It is observed that quasi-interpolation-based method is unconditionally stable, whereas for DQM, the stability has to be checked for a large number of space points. Moreover, for the small number of grid points, DQM gives better results, while for a large number of grid points, quasi-interpolation-based method is better.

Keywords Cubic B-spline functions · Quasi-interpolation · Differential quadrature method · Stability

✉ Sudhir Kumar
skumar4@ma.iitr.ac.in

R. C. Mittal
rcmmmfma@iitr.ac.in

Ram Jiwari
ram03fma@iitr.ac.in

[1] Department of Mathematics, Indian Institute of Technology Roorkee, Roorkee 247667, India

1 Introduction

Consider the forth-order parabolic equation governing the transverse vibrations of a beam,

$$\frac{\partial^2 v}{\partial t^2}+\frac{\partial^4 v}{\partial x^4}=G(x,t),\quad x\in(a,b),\quad t>0, \tag{1}$$

with initial conditions

$$\begin{cases} v(x,0)=f_0(x), & x\in(a,b),\\ v_t(x,0)=f_1(x), & x\in(a,b), \end{cases} \tag{2}$$

and boundary conditions being

$$\begin{cases} v(a,t)=g_a(t), & v(b,t)=g_b(t), & t>0\\ v_{xx}(a,t)=p_a(t), & v_{xx}(b,t)=p_b(t), & t>0 \end{cases} \tag{3}$$

where $v(x, t)$ is the transverse displacement of the beam, t and x are time and space variables, $G(x, t)$ is the dynamic force per unit mass, $f_0(x), f_1(x), g_a(t), g_b(t), p_a(t)$ and $p_b(t)$ are sufficiently smooth functions.

We solve Eq. (1) by rewriting it as a system of two second-order equations, for which we are introducing two new variables ϕ and ψ as

$$\phi=\frac{\partial v}{\partial t},\qquad \psi=\frac{\partial^2 v}{\partial x^2} \tag{4}$$

Now, we get two simultaneous partial differential equations in the following form

$$\begin{cases} \dfrac{\partial \phi}{\partial t}=-\dfrac{\partial^2 \psi}{\partial x^2}+G\\[2ex] \dfrac{\partial \psi}{\partial t}=\dfrac{\partial^2 \phi}{\partial x^2} \end{cases} \tag{5}$$

Equations (2) and (3) now become

$$\phi(x,0) = f(x), \quad \phi(a,t) = a_0(t), \quad \phi(b,t) = b_0(t). \tag{6}$$

$$\psi(x,0) = h(x), \quad \psi(a,t) = a_1(t), \quad \psi(b,t) = b_1(t). \tag{7}$$

Equation (1) has been solved by several authors using finite difference method after splitting into a system of second-order equation [1–6]. Fairweather and Gourlay [7] developed an explicit and implicit scheme which is based on the semi-explicit method of Lees [8]. Mohanty et al. [9] solved a special type of fourth-order parabolic PDE by two-level implicit methods. Mittal and Jain [10] solved Eq. (1) by cubic B-spline collocation method with redefined basis functions. Dehghan and Manafian [11] used the homotopy perturbation method to solve the fourth-order parabolic PDE.

For the approximation of a function and its derivatives, Sablonnière [12, 13] developed a discrete univariate B-spline quasi-interpolation method and verified that the approximation of first derivative of a certain class of functions by this method is better than the approximation by finite difference method. Moreover, he demonstrated that for cubic spline interpolation, the first derivative of certain functions represents the convergence of order $O(h^4)$. Based on this motivation, the research community has tried to implement this technique to develop numerical algorithms for a few partial differential equations. Zhu and Kang [14, 15] solved hyperbolic conservation laws using the quasi-interpolation-based method. Kumar and Baskar [16] developed higher-order numerical schemes for particular one-dimensional Sobolev-type equations by implementing quadratic and cubic B-spline technique for quasi-interpolation and compared the performance of the proposed algorithm in terms of accuracy and the rate of convergence.

Bellman et al. [17] were the first to introduce DQM for the solution of PDEs. Quan and Chang [18] used DQM to develop explicit formulae for approximation of weighting coefficients. There are various types of test functions that have been used in DQM to compute the weighting coefficient, viz. B-spline functions (quadratic, cubic, quintic, etc.), Legendre polynomials, Lagrange interpolation polynomials, sine–cosine function, etc. B-spline functions are piece-wise polynomials and their curves have the property to maintain the smoothness and continuity of higher-order derivatives, and due to the local support property, B-spline functions are commonly used as a test function. Mittal and Jiwari [19, 20] solved nonlinear one-dimensional Berger–Huxley-, Fisher- and Burgers-type equations by DQM, also see [21–24]. Dehghan and Abbaszadeh [25, 26] solved Brusselator reaction–diffusion model and Klein–Gordon Zakharov equations by different methods, also see [27–29].

The main objective of this work is to present a study of CBSQI and DQM for solving fourth-order parabolic PDEs.

2 Univariate B-spline Quasi-interpolants

Let us consider an interval $[a, b]$ with the uniform partition $X_n = \{x_j = a + jh : j = 0, 1, \ldots, n\}$, where $h = (b-a)/n$. Let $B^d(X_n)$ be the spline space of degree d, and let $\{B_j^d : j = 1, 2, \ldots, n+d\}$ form a basis for $B^d(X_n)$, which can be formulated by the de Boor-Cox recursive formula [30]. As we know that support of a B-spline is a subset of the interval $[x_{j-d-1}, x_j]$, we need to add multiple knots at the endpoints in such a way that $x_{-d} = x_{-d+1} = \ldots = x_{-1} = x_0 = a$ and $b = x_n = x_{n+1} = \ldots = x_{n+d}$.

B-spline quasi-interpolant of degree d for a function v has been defined as [13]

$$Q_d v(x) = \sum_{j=1}^{n+d} \mu_j(v) B_j^d(x). \tag{8}$$

Let $\mathbb{P}_n^d$ be the space of polynomial of degree at most d. In general, we impose the condition that quasi-interpolant $Q_d v$ is exact on $\mathbb{P}_n^d$, i.e. $Q_d v = v$ for all $v \in \mathbb{P}_n^d$. The coefficients μ_j are obtained using this condition. Sablonnière [31] used this technique called the *discrete quasi-interpolant*. The main advantage of BSQI is that it is very easy to implement as it has direct construction, i.e., we do not need to solve any system of linear equations. Moreover, it is local, i.e., the value $Q_d v(x)$ depends only on the values of v in a neighborhood of x. We also show that derivatives of B-spline quasi-interpolants are approximated as the derivatives of a corresponding function.

2.1 Cubic B-spline Quasi-Interpolation

For a function v, cubic B-spline quasi-interpolant is defined from Eq. (8) by taking $d = 3$ as

$$Q_3 v(x) = \sum_{j=1}^{n+3} \mu_j(v) B_j^3(x), \tag{9}$$

where nodes are taken to be the same as knots, i.e., $\xi_j = x_j$ $(j = 0, 1, \ldots, n)$ and define the coefficients $\mu_j(v)$ $(j = 1, 2, \ldots, n+3)$

$$\begin{cases} \mu_1(v) = v_0 \qquad \mu_2(v) = \dfrac{1}{18}\Big[7v_0 + 18v_1 - 9v_2 + 2v_3\Big], \\ \mu_j(v) = \dfrac{1}{6}\Big[-v_{j-3} + 8v_{j-2} - v_{j-1}\Big], \quad \text{for } 3 \le j \le (n+1) \\ \mu_{n+2}(v) = \dfrac{1}{18}\Big[2v_{n-3} - 9v_{n-2} + 18v_{n-1} + 7v_n\Big], \quad \mu_{n+3}(v) = v_n, \end{cases} \tag{10}$$

and the corresponding B-spline functions are generated by

$$B_j^3(\xi) = \begin{cases} \dfrac{(\xi - x_{j-4})^3}{(x_{j-3} - x_{j-4})(x_{j-2} - x_{j-4})(x_{j-1} - x_{j-4})}, & \text{if } x_{j-4} < \xi \le x_{j-3}, \\ \dfrac{(\xi - x_{j-4})^2(x_{j-2} - \xi)}{(x_{j-2} - x_{j-4})(x_{j-2} - x_{j-3})(x_{j-1} - x_{j-4})} \\ + \dfrac{(\xi - x_{j-4})(x_{j-1} - \xi)(\xi - x_{j-3})}{(x_{j-1} - x_{j-4})(x_{j-1} - x_{j-3})(x_{j-2} - x_{j-3})} \\ + \dfrac{(x_j - \xi)(\xi - x_{j-3})^2}{(x_j - x_{j-3})(x_{j-1} - x_{j-3})(x_{j-2} - x_{j-3})}, & \text{if } x_{j-3} < \xi \le x_{j-2} \\ \dfrac{(\xi - x_{j-4})(x_{j-1} - \xi)^2}{(x_{j-1} - x_{j-4})(x_{j-1} - x_{j-3})(x_{j-1} - x_{j-2})} \\ + \dfrac{(\xi - x_{j-3})(x_{j-1} - \xi)(x_j - \xi)}{(x_{j-1} - x_{j-3})(x_{j-1} - x_{j-2})(x_j - x_{j-3})} \\ + \dfrac{(x_j - \xi)^2(\xi - x_{j-2})}{(x_j - x_{j-3})(x_j - x_{j-2})(x_{j-1} - x_{j-2})}, & \text{if } x_{j-2} < \xi \le x_{j-1}, \\ \dfrac{(x_j - \xi)^3}{(x_j - x_{j-3})(x_j - x_{j-2})(x_j - x_{j-1})}, & \text{if } x_{j-1} < \xi \le x_j \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

using de Boor-Cox recursive formula [30]
The different B-spline functions are represented in Fig. 1.

The first and the second derivatives of $Q_3 v$ are calculated as

$$(Q_3 v)'(x) = \sum_{j=1}^{n+3} \mu_j(v)(B_j^3)'(x), \tag{12}$$

and

$$(Q_3 v)''(x) = \sum_{j=1}^{n+3} \mu_j(v)(B_j^3)''(x), \tag{13}$$

where $(B_j^3)'$ and $(B_j^3)''$ are obtained from Eq. (11).

$$\left.\begin{aligned} (Q_3 v)'(\xi_0) &= \frac{1}{h}\Big[-\frac{11}{6}v_0 + 3v_1 - \frac{3}{2}v_2 + \frac{1}{3}v_3\Big], \\ (Q_3 v)'(\xi_1) &= \frac{1}{h}\Big[-\frac{1}{3}v_0 - \frac{1}{2}v_1 + v_2 - \frac{1}{6}v_3\Big], \\ (Q_3 v)'(\xi_{n-1}) &= \frac{1}{h}\Big[\frac{1}{6}v_{n-3} - v_{n-2} + \frac{1}{2}v_{n-1} + \frac{1}{3}v_n\Big], \\ (Q_v)'(\xi_n) &= \frac{1}{h}\Big[-\frac{1}{3}v_{n-3} + \frac{3}{2}v_{n-2} - 3v_{n-1} + \frac{11}{6}v_n\Big], \end{aligned}\right\} \tag{14}$$

and

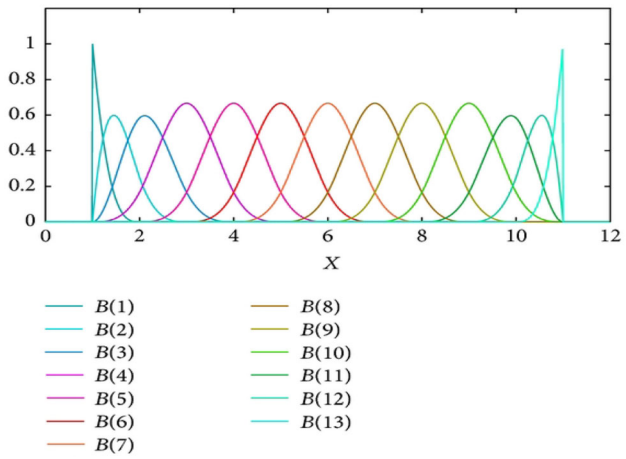

Fig. 1 Cubic B-spline functions for the knot $X_{10} = (1, 1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 11, 11)$, $B_j^3, j = 1, 2, \ldots, 13$

$$(Q_3v)'(\xi_j) = \frac{1}{h}\left[\frac{1}{12}v_{j-2} - \frac{2}{3}v_{j-1} + \frac{2}{3}v_{j+1} - \frac{1}{12}v_{j+2}\right], \quad 2 \le j \le (n-2). \tag{15}$$

The approximation of v' can be written in terms of matrix form as

$$(Q_3v)' = \frac{1}{h}D_3^{(1)}v, \tag{16}$$

where $D_3^{(1)}$ is the $(n+1) \times (n+1)$ coefficient matrix that is obtained from Eqs. (14)–(15) and $v = (v_0, v_1, \ldots, v_n)^T$.

For the second derivative, we have

$$\left.\begin{aligned}
(Q_3v)''(\xi_0) &= \frac{1}{h^2}\left[2v_0 - 5v_1 + 4v_2 - v_3\right],\\
(Q_3v)''(\xi_1) &= \frac{1}{h^2}\left[v_0 - 2v_1 + v_2\right],\\
(Q_3v)''(\xi_{n-1}) &= \frac{1}{h^2}\left[v_{n-2} - 2v_{n-1} + v_n\right],\\
(Q_3v)''(\xi_n) &= \frac{1}{h^2}\left[-v_{n-3} + 4v_{n-2} - 5v_{n-1} + 2v_n\right],
\end{aligned}\right\} \tag{17}$$

and

$$(Q_3v)''(\xi_j) = \frac{1}{h^2}\left[-\frac{1}{6}v_{j-2} + \frac{5}{3}v_{j-1} - 3v_j + \frac{5}{3}v_{j+1} - \frac{1}{6}v_{j+2}\right], \quad 2 \le j \le (n-2). \tag{18}$$

Similarly, we write the above expressions in terms of matrix form as

$$(Q_3v)'' = \frac{1}{h^2}D_3^{(2)}v, \tag{19}$$

where $D_3^{(2)}$ is the $(n+1) \times (n+1)$ coefficient matrix that is obtained from Eqs. (17)–(18).

3 Description of Cubic B-spline Quasi-interpolation Method (CBSQI)

Now we implement the CBSQI method, discretizing the time derivative as forward difference scheme and for space derivative applying θ-weighted scheme in Eq. (5), where $0 \le \theta \le 1$, giving

$$\frac{\phi^{m+1} - \phi^m}{\Delta t} = -\left[\theta\psi_{xx}^{m+1} + (1-\theta)\psi_{xx}^m\right] + \left[\theta G^{m+1} + (1-\theta)G^m\right], \tag{20}$$

$$\frac{\psi^{m+1} - \psi^m}{\Delta t} = \left[\theta\phi_{xx}^{m+1} + (1-\theta)\phi_{xx}^m\right]. \tag{21}$$

$$\phi^{m+1} + \frac{\Delta t\theta}{h^2}D_3^{(2)}\psi^{m+1} = \phi^m - \frac{\Delta t(1-\theta)}{h^2}D_3^{(2)}\psi^m + \Delta t\left[\theta G^{m+1} + (1-\theta)G^m\right], \tag{22}$$

$$\psi^{m+1} - \frac{\Delta t\theta}{h^2}D_3^{(2)}\phi^{m+1} = \psi^m + \frac{\Delta t(1-\theta)}{h^2}D_3^{(2)}\phi^m \tag{23}$$

Although Eqs. (22) and (23) are valid for all $\theta \in [0, 1]$, we will use $\theta = \frac{1}{2}$(the famous Crank–Nicolson scheme)

$$\phi^{m+1} + \frac{\Delta t}{2h^2}D_3^{(2)}\psi^{m+1} = \phi^m - \frac{\Delta t}{2h^2}D_3^{(2)}\psi^m + \frac{\Delta t}{2}\left[G^{m+1} + G^m\right], \tag{24}$$

$$\psi^{m+1} - \frac{\Delta t}{2h^2}D_3^{(2)}\phi^{m+1} = \psi^m + \frac{\Delta t}{2h^2}D_3^{(2)}\phi^m \tag{25}$$

Let $\mu = \dfrac{\Delta t}{2h^2}$

$$\phi^{m+1} + \mu D_3^{(2)}\psi^{m+1} = \phi^m - \mu D_3^{(2)}\psi^m + \frac{\Delta t}{2}\left[G^{m+1} + G^m\right] \tag{26}$$

$$\psi^{m+1} - \mu D_3^{(2)}\phi^{m+1} = \psi^m + \mu D_3^{(2)}\phi^m \tag{27}$$

$$\begin{bmatrix} \mu D_3^{(2)} & I \\ I & -\mu D_3^{(2)} \end{bmatrix}\begin{bmatrix} \psi^{m+1} \\ \phi^{m+1} \end{bmatrix} = \begin{bmatrix} -\mu D_3^{(2)} & I \\ I & \mu D_3^{(2)} \end{bmatrix}\begin{bmatrix} \psi^m \\ \phi^m \end{bmatrix} + \frac{\Delta t}{2}\begin{bmatrix} G^{m+1} + G^m \\ \mathbf{0} \end{bmatrix} \tag{28}$$

where

$$D_3^{(2)} = \begin{pmatrix} 2 & -5 & 4 & -1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \frac{-1}{6} & \frac{5}{3} & -3 & \frac{5}{3} & \frac{-1}{6} & 0 & \cdots & 0 & 0 \\ 0 & \frac{-1}{6} & \frac{5}{3} & -3 & \frac{5}{3} & \frac{-1}{6} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \frac{-1}{6} & \frac{5}{3} & -3 & \frac{5}{3} & \frac{-1}{6} & 0 \\ 0 & 0 & \cdots & 0 & \frac{-1}{6} & \frac{5}{3} & -3 & \frac{5}{3} & \frac{-1}{6} \\ 0 & 0 & \cdots & 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & \cdots & 0 & 0 & -1 & 4 & -5 & 2 \end{pmatrix}_{(n+1)\times(n+1)} \tag{29}$$

where $D_3^{(2)}$ and I (identity) are $(n+1)\times(n+1)$ matrices and $\boldsymbol{\phi}^m = (\phi_1^m, \phi_2^m, \ldots, \phi_{n+1}^m)^T$ and $\boldsymbol{\psi}^m = (\psi_1^m, \psi_2^m, \ldots, \psi_{n+1}^m)^T$ are the column vectors. When $m = 0$, the vectors $\boldsymbol{\phi}^0$ and $\boldsymbol{\psi}^0$ are obtained from the initial conditions and solutions of Eq. (5) at time level $t = (m+1)\Delta t$ are calculated by solving the linear system Eq. (28). After calculating the value of ψ at each time level, we again apply the CBSQI on Eq. (4) to get the final result.

4 Stability of CBSQI

Since stability does not depend on $G(x, t)$, so in stability discussion we ignore $G(x, t)$. By using the coefficients of CBSQI from Eqs. (17)–(18) to approximate the space derivative, Eqs. (26)–(27) are written for internal nodes as

$$\begin{aligned} &\phi_i^{m+1} + \mu\Big[-\frac{1}{6}\psi_{i-2}^{m+1} + \frac{5}{3}\psi_{i-1}^{m+1} - 3\psi_i^{m+1} + \frac{5}{3}\psi_{i+1}^{m+1} - \frac{1}{6}\psi_{i+2}^{m+1}\Big] \\ &= \phi_i^m - \mu[-\frac{1}{6}\psi_{i-2}^m + \frac{5}{3}\psi_{i-1}^m - 3\psi_i^m + \frac{5}{3}\psi_{i+1}^m \\ &\quad -\frac{1}{6}\psi_{i+2}^m\Big], \text{for} i = 2,\ldots,n-1 \end{aligned} \tag{30}$$

$$\begin{aligned} &\psi_i^{m+1} - \mu\Big[-\frac{1}{6}\phi_{i-2}^{m+1} + \frac{5}{3}\phi_{i-1}^{m+1} - 3\phi_i^{m+1} + \frac{5}{3}\phi_{i+1}^{m+1} - \frac{1}{6}\phi_{i+2}^{m+1}\Big] \\ &= \psi_i^m + \mu\Big[-\frac{1}{6}\phi_{i-2}^m + \frac{5}{3}\phi_{i-1}^m - 3\phi_i^m + \frac{5}{3}\phi_{i+1}^m \\ &\quad -\frac{1}{6}\phi_{i+2}^m\Big], \text{for} i = 2,\ldots,n-1 \end{aligned} \tag{31}$$

$$\phi^{m+1} + \mu T\psi^{m+1} = \phi^m - \mu T\psi^m \tag{32}$$

$$-\mu T\phi^{m+1} + \psi^{m+1} = \psi^m + \mu T\phi^m \tag{33}$$

where

$$T = \begin{pmatrix} -3 & \frac{5}{3} & \frac{-1}{6} & 0 & 0 & \cdots & 0 & 0 \\ \frac{5}{3} & -3 & \frac{5}{3} & \frac{-1}{6} & 0 & \cdots & 0 & 0 \\ \frac{-1}{6} & \frac{5}{3} & -3 & \frac{5}{3} & \frac{-1}{6} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \frac{-1}{6} & \frac{5}{3} & -3 & \frac{5}{3} & \frac{-1}{6} \\ 0 & 0 & \cdots & 0 & \frac{-1}{6} & \frac{5}{3} & -3 & \frac{5}{3} \\ 0 & 0 & \cdots & 0 & 0 & \frac{-1}{6} & \frac{5}{3} & -3 \end{pmatrix}_{(n-3)\times(n-3)} \tag{34}$$

Table 1 Cubic B-splines and its derivatives at knots

x	x_{j-2}	x_{j-1}	x_j	x_{j+1}	x_{j+2}
$B_j^3(x)$	0	1	4	1	0
$B_j^3(x)'$	0	$\frac{3}{h}$	0	$-\frac{3}{h}$	0
$B_j^3(x)''$	0	$\frac{6}{h^2}$	$-\frac{12}{h^2}$	$\frac{6}{h^2}$	0

We write above equation as

$$\begin{bmatrix} I & \mu T \\ -\mu T & I \end{bmatrix}_{2(n-3)\times 2(n-3)} \begin{bmatrix} \phi^{m+1} \\ \psi^{m+1} \end{bmatrix} = \begin{bmatrix} I & -\mu T \\ \mu T & I \end{bmatrix}_{2(n-3)\times 2(n-3)} \begin{bmatrix} \phi^{m} \\ \psi^{m} \end{bmatrix} \tag{35}$$

$$\begin{bmatrix} \phi^{m+1} \\ \psi^{m+1} \end{bmatrix} = (I+\mu^2T^2)^{-1} \begin{bmatrix} I & -\mu T \\ \mu T & I \end{bmatrix} \begin{bmatrix} I & -\mu T \\ \mu T & I \end{bmatrix} \begin{bmatrix} \phi^{m} \\ \psi^{m} \end{bmatrix} \tag{36}$$

$$\begin{bmatrix} \phi^{m+1} \\ \psi^{m+1} \end{bmatrix} = (I+\mu^2T^2)^{-1} \begin{bmatrix} I-\mu^2T^2 & -2\mu T \\ 2\mu T & I-\mu^2T^2 \end{bmatrix} \begin{bmatrix} \phi^{m} \\ \psi^{m} \end{bmatrix} \tag{37}$$

$$W^{m+1} = \mathbb{M} W^m \tag{38}$$

where

$$W^{m+1} = \begin{bmatrix} \phi^{m+1} \\ \psi^{m+1} \end{bmatrix}, \mathbb{M} = (I+\mu^2T^2)^{-1} \begin{bmatrix} I-\mu^2T^2 & -2\mu T \\ 2\mu T & I-\mu^2T^2 \end{bmatrix}$$

If the eigenvalues of T be $\tau_i, i = 1,2,\ldots,n-1,$, then the eigenvalues λ_i of $\mathbb{M}$ are obtained as

$$\lambda_i = \frac{1-\mu^2\tau_i^2}{1+\mu^2\tau_i^2} \pm 2j\frac{\mu\tau_i}{1+\mu^2\tau_i^2}, \qquad j = \sqrt{-1} \tag{39}$$

The modulus of the above expression is equal to unity, which shows that the method is unconditionally stable (Table 1).

5 Modified Cubic B-spline Differential Quadrature Method

In DQM, the approximation of the derivatives of a certain function is achieved by writing it as the weighted sum of its values at discrete points over the considered domain. The remaining work is to calculate the weighting coefficients. For this, we consider uniformly distributed n knots: $a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b$ such that $x_{i+1} - x_i = h$. For a given function $v(x, t)$, first- and second-order spatial derivatives at any node x_i for $i = 0, 1, \ldots, n$ are approximated by

$$v_x(x_i,t) = \sum_{j=0}^{n} \alpha_{ij} v(x_j,t), \qquad \text{for } i = 0,1,\ldots,n, \tag{40}$$

$$v_{xx}(x_i,t) = \sum_{j=0}^{n} \beta_{ij} v(x_j,t), \qquad \text{for } i = 0,1,\ldots,n. \tag{41}$$

where α_{ij} and β_{ij} are the weighting coefficients of the first- and second-order derivatives with respect to space variable. We have cubic B-spline function from Eq. (11), from which set $\{B_{-1}^3(x), B_0^3(x), \ldots, B_n^3(x), B_{n+1}^3(x)\}$ forms a basis over the considered domain. By using these functions, we define the modified cubic B-spline functions at any node as

$$\left.\begin{aligned} \check{B}_0(x) &= B_0^3(x) + 2B_{-1}^3(x), \\ \check{B}_1(x) &= B_1^3(x) - B_{-1}^3(x), \\ \check{B}_i(x) &= B_i^3(x), \qquad i = 2,4,\ldots,n-2, \\ \check{B}_{n-1}(x) &= B_{n-1}^3(x) - B_{n+1}^3(x), \\ \check{B}_n(x) &= B_n^3(x) + 2B_{n+1}^3(x), \end{aligned}\right\} \tag{42}$$

where set $\{\check{B}_0(x), \check{B}_1(x), \ldots, \check{B}_{n-1}(x), \check{B}_n(x)\}$ forms a basis over the considered interval. The values of cubic B-splines and its derivatives at the nodes are presented in Table 1.

5.1 Computation of the Weighting Coefficients

The first-order derivative is approximated as

$$\check{B}_k{}'(x_i) = \sum_{j=0}^{n} \omega_{ij}^{(1)} \check{B}_k(x_j), \quad i = 0,1,\ldots,n \text{ and } k = 0,1,\ldots,n \tag{43}$$

At the first knot x_0, the approximation is given as

$$\check{B}_k{}'(x_0) = \sum_{j=0}^{n} \omega_{0j}^{(1)} \check{B}_k(x_j), \quad k = 0,1,\ldots,n. \tag{44}$$

For $x = x_0$, the value of $\check{B}_k'(x_0)$ is given by $6/h$ at x_1 knot and $-6/h$ at x_0 knot.

This results in a tridiagonal system of equations as

$$\begin{pmatrix} 6 & 1 & 0 & & & \\ 0 & 4 & 1 & 0 & & \\ & & 1 & 4 & 1 & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ & & 1 & 4 & 1 & \\ & & 0 & 1 & 4 & 0 \\ & & & 0 & 1 & 6 \end{pmatrix} \begin{pmatrix} \omega_{00}^{(1)} \\ \omega_{01}^{(1)} \\ . \\ . \\ . \\ \omega_{0n}^{(1)} \end{pmatrix} = \begin{pmatrix} -\frac{6}{h} \\ \frac{6}{h} \\ . \\ . \\ . \\ 0 \end{pmatrix} \tag{45}$$

We note that the above coefficient matrix is nonsingular. So to solve the above system, we apply Thomas algorithm whose solution gives us the coefficients $\omega_{00}^{(1)}, \omega_{01}^{(1)}, \ldots, \omega_{0n}^{(1)}$.

Similarly, for second knot x_1, the approximation is given as

$$\check{B}_k'(x_1) = \sum_{j=0}^{n} \omega_{1j}^{(1)} \check{B}_k(x_j), \quad k = 0, 1, \ldots, n \tag{46}$$

which again results in a tridiagonal system of equations as follows

$$\begin{pmatrix} 6 & 1 & 0 & & & \\ 0 & 4 & 1 & 0 & & \\ & & 1 & 4 & 1 & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ & & 1 & 4 & 1 & \\ & & 0 & 1 & 4 & 0 \\ & & & 0 & 1 & 6 \end{pmatrix} \begin{pmatrix} \omega_{10}^{(1)} \\ \omega_{11}^{(1)} \\ . \\ . \\ . \\ \omega_{1n}^{(1)} \end{pmatrix} = \begin{pmatrix} \frac{3}{h} \\ 0 \\ -\frac{3}{h} \\ . \\ . \\ 0 \end{pmatrix} \tag{47}$$

The solution of the above system provides the coefficients $\omega_{10}^{(1)}, \omega_{11}^{(1)}, \ldots, \omega_{1n}^{(1)}$. In the same way, the weighting coefficients corresponding to $x_i, i = 2, 3, \ldots, n-1$ are determined. Finally, for the last knot, $\check{B}_k'(x_n)$ is given by $6/h$ at x_n and $-6/h$ at x_{n-1}.

$$\check{B}_k'(x_n) = \sum_{j=0}^{n} \omega_{nj}^{(1)} \check{B}_k(x_j), \quad k = 0, 1, \ldots, n \tag{48}$$

$$\begin{pmatrix} 6 & 1 & 0 & & & \\ 0 & 4 & 1 & 0 & & \\ & & 1 & 4 & 1 & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ & & 1 & 4 & 1 & \\ & & 0 & 1 & 4 & 0 \\ & & & 0 & 1 & 6 \end{pmatrix} \begin{pmatrix} \omega_{n0}^{(1)} \\ \omega_{n1}^{(1)} \\ . \\ . \\ . \\ \omega_{nn}^{(1)} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ . \\ . \\ -\frac{6}{h} \\ \frac{6}{h} \end{pmatrix} \tag{49}$$

for which solution provides the coefficients $\omega_{n0}^{(1)}, \omega_{n1}^{(1)}, \ldots, \omega_{nn}^{(1)}$. Thus, we have calculated the first-order weighting coefficient $\omega_{ij}^{(1)}$ of B-spline functions for $0 \le i, j \le n$.

In the same way, the weighting coefficient $\omega_{ij}^{(2)}, 0 \le i, j \le n$ for the second-order partial derivative, is determined. Second- or higher-order weighting coefficients are computed by using Shu recursion formula [32]:

$$\omega_{ij}^{(k)} = k\left[\omega_{ij}^{(1)} \omega_{ii}^{(k-1)} - \frac{\omega_{ij}^{(k-1)}}{(x_i - x_j)}\right], \tag{50}$$

$$\omega_{ii}^{(k)} = -\sum_{j=0, j\neq i}^{n} \omega_{ij}^{(k)}, \; i = j \text{ and } i = 0, 1, \ldots, n \tag{51}$$

6 Implementation of Differential Quadrature Method (DQM)

Applying the DQM to Eq. (5) , we get

$$\frac{\mathrm{d}\phi(x_i, t)}{\mathrm{d}t} = -\sum_{j=1}^{n-1} \omega_{ij}^{(2)} \psi(x_i, t) + G(x_i, t), \quad \text{for } i = 1, \ldots, n-1, \tag{52}$$

$$\frac{\mathrm{d}\psi(x_i, t)}{\mathrm{d}t} = \sum_{j=1}^{n-1} \omega_{ij}^{(2)} \phi(x_i, t), \quad \text{for } i = 1, \ldots, n-1, \tag{53}$$

with initial conditions and boundary conditions Eqs. (6) and (7). Now the above system is written as

$$\frac{\mathrm{d}\Omega}{\mathrm{d}t} = \mathbb{P}\Omega + \mathbb{Q} \tag{54}$$

where

$$\mathbb{P} = \begin{bmatrix} 0 & -A \\ A & 0 \end{bmatrix}, \; \Omega = \begin{bmatrix} \phi \\ \psi \end{bmatrix} \text{and } \mathbb{Q} = \begin{bmatrix} G_1 \\ 0 \end{bmatrix} \tag{55}$$

where A is a matrix of the weighting coefficients $\omega_{ij}^{(2)}$ and $\mathbb{Q}$ contains boundary and other values. Then, the above system of ODE is integrated w.r.t. time deploying an appropriate method. Here, strong stability-preserving fourth-order RK method is preferred for its inherent advantages such as correctness of solution, numerical stability and compact memory requirements.

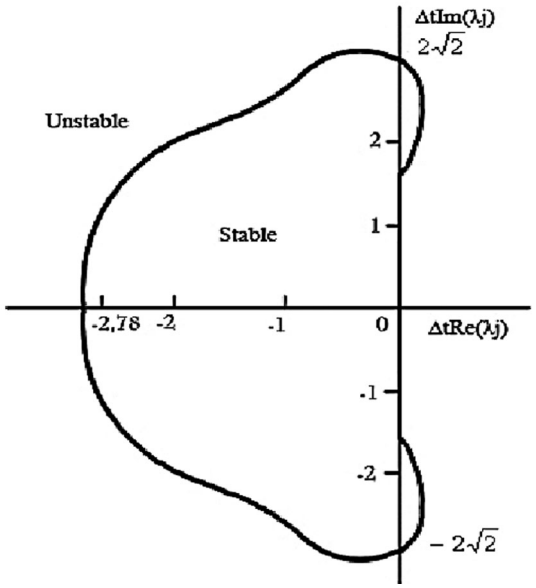

Fig. 2 Stability region

7 Stability of DQM

From Eq. (54), we have system

$$\frac{\mathrm{d}\Omega}{\mathrm{d}t} = \mathbb{P}\Omega + \mathbb{Q} \tag{56}$$

where $\Omega = [\phi\ \psi]^T = [\phi_2, \phi_3, \ldots, \phi_{n-1}\ \ \psi_2, \psi_3, \ldots, \psi_{n-1}]^T$ is the solution vector at the internal nodes, $\mathbb{P}$ is the coefficient matrix and the vector $\mathbb{Q}$ representing the boundary and other values.

Assume that λ_i is the eigenvalue of $\mathbb{P}$. Asymptotically, for the stable solution of Ω, we must have

1. $-2.78 < \Delta t\lambda_i < 0$, if eigenvalues are real.
2. $-2\sqrt{2} < \Delta t\lambda_i < 2\sqrt{2}$, if eigenvalues have complex part only.
3. $\Delta t\lambda_i$ should be in a region as shown in Fig. 2, if eigenvalues are complex.

Eigenvalues of $\mathbb{P}$ depend upon the eigenvalues of A, which are found to be within the stability region. Similarly, we can check the stability of nonlinear problems.

8 Numerical Experiments

This section presents the results obtained by CBSQI and DQM in graphical and tabular forms with the brief description. The accuracy and efficiency of the proposed method are calculated for four test problems by maximum absolute error norm, which is defined as follows.

$$L_\infty = \|v^{\text{exact}} - v^{\text{cal}}\| = \max_i \mid v_i^{\text{exact}} - v_i^{\text{cal}} \mid \tag{57}$$

where v_i^{exact} and v_i^{cal} denote the exact and calculated solutions at knot x_i, respectively.

The convergence rate of the DQM is to be evaluated, which is obtained by L_∞ error norm. The following formula has been used to compute the order of convergence:

$$\text{Order} = \frac{\log\left(E(N_1)/E(N_2)\right)}{\log(N_1/N_2)} \tag{58}$$

where $E(N_1)$ is the error and N_1 is the count of partitions.

We have calculated the rate of convergence of the CBSQI method for Problem 1. Figure 3 shows that CBSQI provides the second-order approximation.

Problem 1 Consider a fourth-order nonhomogeneous PDE

$$\frac{\partial^2 v}{\partial t^2} + \frac{\partial^4 v}{\partial x^4} = (\pi^4 - 1)\sin \pi x \cos t, \quad x \in [0, 1], \quad t > 0$$

along with initial conditions

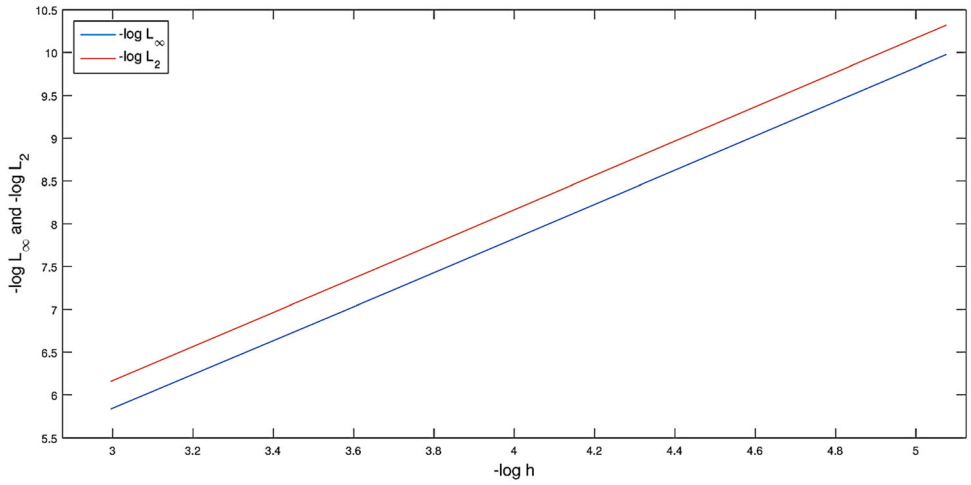

Fig. 3 Log–log plot between errors and space step

Table 2 Maximum absolute errors in $v(x, t)$ and $v_{xx}(x, t)$ for Problem 1

Methods	Time	Parameters	v	v_{xx}
Method 1	$t = 0.02$	$n = 20$, $\Delta t = 0.0001$	2.0567E−03	4.9677E−04
CBSQI	$t = 0.02$	$n = 40$, $\Delta t = 0.0001$	5.2226E−04	1.0257E−04
	$t = 0.02$	$n = 60$, $\Delta t = 0.0001$	2.3262E−04	4.4801E−05
	$t = 0.02$	$n = 90$, $\Delta t = 0.0001$	1.0347E−04	1.9886E−05
	$t = 0.02$	$n = 180$, $\Delta t = 0.0001$	2.5879E−05	4.9653E−06
	$t = 0.02$	$n = 270$, $\Delta t = 0.0001$	1.1503E−05	2.2063E−06
	$t = 0.1$	$n = 20$, $\Delta t = 0.0001$	9.9200E−03	9.1832E−03
	$t = 0.1$	$n = 40$, $\Delta t = 0.0001$	7.4170E−04	2.2967E−03
	$t = 0.1$	$n = 60$, $\Delta t = 0.0001$	3.3036E−04	1.0211E−03
	$t = 0.1$	$n = 90$, $\Delta t = 0.0001$	1.4695E−04	4.5386E−04
	$t = 0.1$	$n = 180$, $\Delta t = 0.0001$	3.6752E−05	1.1348E−04
	$t = 0.1$	$n = 270$, $\Delta t = 0.0001$	1.6335E−05	5.0438E−05
Method 2	$t = 0.02$	$n = 20$, $\Delta t = 0.0001$	2.0569E−03	1.72651E−03
DQM	$t = 0.02$	$n = 40$, $\Delta t = 0.0001$	5.15044E−04	1.81437E−04
	$t = 0.02$	$n = 60$, $\Delta t = 0.0001$	2.29299E−04	6.6936E−05
	$t = 0.1$	$n = 20$, $\Delta t = 0.0001$	2.13265E−03	1.78313E−03
	$t = 0.1$	$n = 40$, $\Delta t = 0.0001$	5.32746E−04	3.76701E−04
	$t = 0.1$	$n = 60$, $\Delta t = 0.0001$	2.32697E−04	2.03133E−04

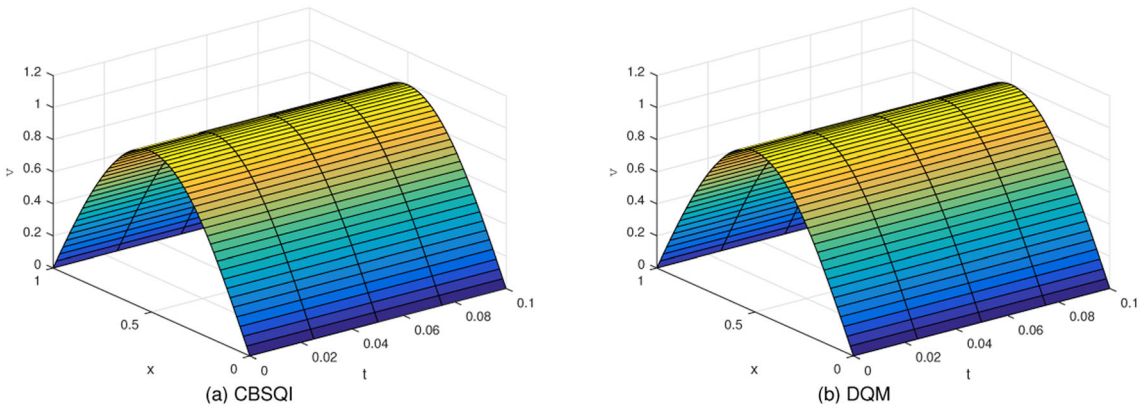

Fig. 4 Displacement for $t \leq 0.1$ ($n = 60$, $\Delta t = 0.0001$)

$$v(x, 0) = \sin \pi x, \quad v_t(x, 0) = 0, \quad x \in [0, 1]$$

and boundary conditions

$$v(0, t) = v(1, t) = v_{xx}(0, t) = v_{xx}(1, t) = 0, \quad t \geq 0.$$

The analytical solution is $v(x, t) = \sin \pi x \cos t$. From Eq. (4), initial and boundary conditions are derived as

$$\phi(x, 0) = 0, \quad \phi(0, t) = \phi(1, t) = 0.$$
$$\psi(x, 0) = -\pi^2 \sin \pi x, \quad \psi(0, t) = \psi(1, t) = 0.$$

The computed results obtained by both methods and analytical solution are compared in Table 2.

In Table 2, displacement $v(x, t)$ and bending moment $v_{xx}(x, t)$ are computed for different values of $t = 0.02$ and 0.1 for each $n = 20, 40, 60$ and $\Delta t = 0.0001$. We observe that computed results by DQM for $n = 20, 40, 60$ are better than CBSQI. But instead of this, we also observe that CBSQI produces good results for large $n = 90, 180, 270$, while DQM becomes unstable for such large n.

Figure 4 depicts the computed numerical results for $n = 60$, $\Delta t = 0.0001$ at $t = 0.1$. In Table 3, we compare our results with Mittal and Jain [10], and it is clear that CBSQI method gives good results.

Table 3 Maximum absolute errors in $v(x, t)$ for Problem 1

Methods	Time	Parameters	$x = 0.1$	$x = 0.2$	$x = 0.3$	$x = 0.4$	$x = 0.5$
Method 1	$t = 0.02$	$n = 90$, $\Delta t = 0.005$	3.20E−05	6.08E−05	8.37E−05	9.84E−05	1.03E−04
CBSQI	$t = 0.05$	$n = 90$, $\Delta t = 0.005$	3.58E−05	6.82E−05	9.39E−05	1.10E−04	1.16E−04
	$t = 1.0$	$n = 90$, $\Delta t = 0.005$	6.31E−05	1.20E−04	1.65E−04	1.94E−04	2.04E−04
	$t = 0.02$	$n = 180$, $\Delta t = 0.005$	8.00E−06	1.52E−05	2.09E−05	2.46E−05	2.59E−05
	$t = 0.05$	$n = 180$, $\Delta t = 0.005$	8.97E−06	1.70E−05	2.35E−05	2.76E−05	2.90E−05
	$t = 1.0$	$n = 180$, $\Delta t = 0.005$	1.58E−05	3.00E−05	4.13E−05	4.86E−05	5.10E−05
	$t = 0.02$	$n = 270$, $\Delta t = 0.005$	3.55E−06	6.76E−06	9.30E−06	1.09E−05	1.15E−05
	$t = 0.05$	$n = 270$, $\Delta t = 0.005$	3.99E−06	7.58E−06	1.04E−05	1.23E−05	1.29E−05
	$t = 1.0$	$n = 270$, $\Delta t = 0.005$	7.00E−06	1.33E−05	1.83E−05	2.15E−05	2.27E−05
Mittal and Jain [10]	$t = 0.02$	$n = 90$, $\Delta t = 0.005$	3.20E−05	6.08E−05	8.37E−05	9.84E−05	1.04E−04
	$t = 0.05$	$n = 90$, $\Delta t = 0.005$	3.59E−05	6.83E−05	9.39E−05	1.10E−04	1.16E−04
	$t = 1.0$	$n = 90$, $\Delta t = 0.005$	6.32E−05	1.20E−04	1.65E−04	1.94E−04	2.04E−04
	$t = 0.02$	$n = 180$, $\Delta t = 0.005$	8.00E−06	1.52E−05	2.09E−05	2.46E−05	2.59E−05
	$t = 0.05$	$n = 180$, $\Delta t = 0.005$	8.97E−06	1.71E−05	2.35E−05	2.76E−05	2.90E−05
	$t = 1.0$	$n = 180$, $\Delta t = 0.005$	1.58E−05	3.21E−05	4.13E−05	4.86E−05	5.11E−05
	$t = 0.02$	$n = 270$, $\Delta t = 0.005$	3.55E−06	6.76E−06	9.30E−06	1.09E−05	1.15E−05
	$t = 0.05$	$n = 270$, $\Delta t = 0.005$	3.99E−06	7.58E−06	1.04E−05	1.23E−05	1.29E−05
	$t = 1.0$	$n = 270$, $\Delta t = 0.005$	7.00E−06	1.33E−05	1.83E−05	2.16E−05	2.27E−05

Problem 2 Consider the singularly perturbed problem of the form:

$$\frac{\partial^2 v}{\partial t^2} + \epsilon \frac{\partial^4 v}{\partial x^4} = f(x,t), \quad 0 < \epsilon << 1, \quad x \in (0,1), \quad t > 0$$

The analytical solution is $v(x,t) = \mathrm{e}^{-\epsilon\pi^2 t} \sin \pi x$. From this, we get initial conditions

$$v(x,0) = \sin \pi x, \quad v_t(x,0) = -\epsilon\pi^2 \sin \pi x, \quad x \in [0,1]$$

and boundary conditions

$$v(0,t) = v(1,t) = v_{xx}(0,t) = v_{xx}(1,t) = 0, \quad t \geq 0.$$

From Eq. (4), initial and boundary conditions are derived as

$$\phi(x,0) = -\epsilon\pi^2 \sin \pi x, \quad \phi(0,t) = 0 = \phi(1,t) = 0.$$
$$\psi(x,0) = -\pi^2 \sin \pi x, \quad \psi(0,t) = \psi(1,t) = 0.$$

The computed results obtained by both methods and analytical solution are compared in Table 4.

Table 4 Maximum absolute errors in $v(x, t)$ and $v_{xx}(x,t)$ for Problem 2

Methods	h		$\epsilon = 0.1$	$\epsilon = 0.01$	$\epsilon = 0.001$
Method 1	1/8	v	1.45,943E−02	1.43,638E−02	1.08,088E−02
CBSQI		v_{xx}	1.08,208E−01	5.68,259E−02	1.75,591E−02
	1/16	v	4.58,819E−03	4.41,718E−03	3.25,827E−03
		v_{xx}	3.41,216E−02	1.63,861E−02	2.28,884E−03
	1/32	v	1.20,114E−03	1.13,281E−03	8.3672E−04
		v_{xx}	8.92,814E−03	4.05,920E−03	4.7459E−04
Method 2	1/8	v	-	1.28,385E−02	1.28,672E−02
DQM		v_{xx}	–	2.90,029E−02	2.10,645E−02
	1/16	v	–	2.91,276E−03	3.19,314E−03
		v_{xx}	–	3.91,293E−03	2.50,244E−03
	1/32	v	–	6.86288E−04	7.94947E−04
		u_{xx}	–	7.53403E−04	2.55585E−04

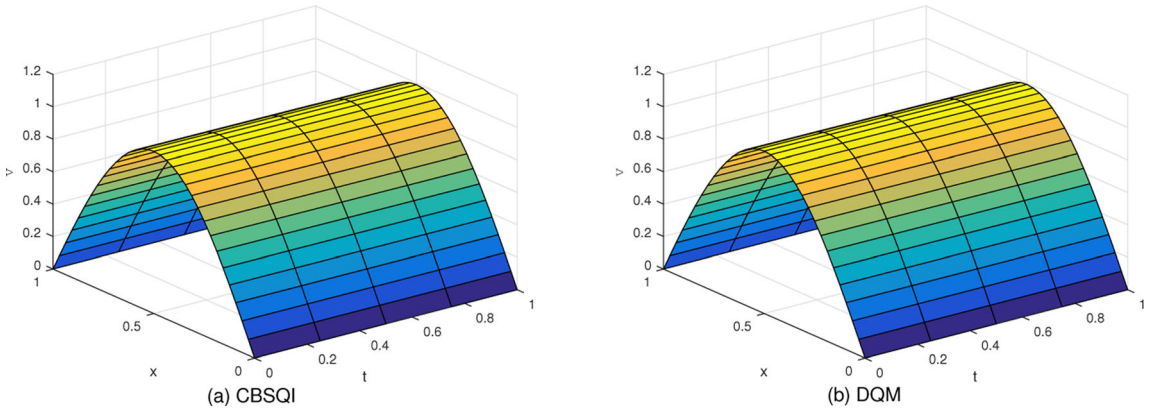

Fig. 5 Displacement for $t \leq 1$ ($n = 32$, $\Delta t = 0.0015625$, $\epsilon = 0.001$)

In Table 4, using $h = 1/8, 1/16, 1/32$ and corresponding $\Delta t = 0.025, 0.00625, 0.0015625$, we compute displacement $v(x, t)$ and bending moment $v_{xx}(x, t)$ for different values of $\epsilon = 0.1, 0.01, 0.001$ and time level $t = 1$ by applying both methods. We found that for $\epsilon = 0.01, 0.001$ computed results by DQM are better than CBSQI. But instead of this, we also observe that CBSQI produces good results for large $\epsilon = 0.1$, while DQM becomes unstable.

Figure 5 depicts the computed numerical results for $n = 32$, $\Delta t = 0.0015625$ at $t = 1$.

Problem 3 Consider a fourth-order nonhomogeneous PDE

$$\frac{\partial^2 v}{\partial t^2} + \frac{\partial^4 v}{\partial x^4} = [24 - x^2(1-x)^2]\cos t, \quad x \in [0,1], \quad t > 0$$

along with initial conditions

$$v(x,0) = x^2(1-x)^2, \quad v_t(x,0) = 0, \qquad x \in [0,1]$$

and boundary conditions

$$v(0,t) = v(1,t) = 0, \quad v_{xx}(0,t) = v_{xx}(1,t) = 2\cos t, \quad t \geq 0$$

The analytical solution is $v(x,t) = x^2(1-x)^2 \cos t$. From Eq. (4), initial and boundary conditions are derived as

Table 5 Maximum absolute errors in $v(x, t)$ and $v_{xx}(x, t)$ for Problem 3

Methods	Time	Parameters	v	v_{xx}
Method 1	$t = 0.02$	$n = 20$, $\Delta t = 0.0001$	5.9220E−04	1.0591E−06
CBSQI	$t = 0.02$	$n = 40$, $\Delta t = 0.0001$	1.5420E−04	2.7463E−07
	$t = 0.02$	$n = 60$, $\Delta t = 0.0001$	6.9039E−05	1.2404E−07
	$t = 0.02$	$n = 90$, $\Delta t = 0.0001$	3.0784E−05	5.5517E−08
	$t = 0.02$	$n = 180$, $\Delta t = 0.0001$	7.7110E−06	1.3939E−08
	$t = 0.02$	$n = 270$, $\Delta t = 0.0001$	3.4284E−06	6.1999E−09
	$t = 0.1$	$n = 20$, $\Delta t = 0.0001$	5.9201E−04	2.7240E−05
	$t = 0.1$	$n = 40$, $\Delta t = 0.0001$	1.5416E−04	7.1033E−06
	$t = 0.1$	$n = 60$, $\Delta t = 0.0001$	6.9021E−05	3.1811E−06
	$t = 0.1$	$n = 90$, $\Delta t = 0.0001$	3.0784E−05	5.5517E−08
	$t = 0.1$	$n = 180$, $\Delta t = 0.0001$	7.7110E−06	1.3939E−08
	$t = 0.1$	$n = 270$, $\Delta t = 0.0001$	3.4284E−06	6.1999E−09
Method 2	$t = 0.02$	$n = 20$, $\Delta t = 0.0001$	6.13641E−04	2.05421E−03
DQM	$t = 0.02$	$n = 40$, $\Delta t = 0.0001$	1.39918E−04	6.92248E−04
	$t = 0.02$	$n = 60$, $\Delta t = 0.0001$	5.23068E−05	6.27756E−04
	$t = 0.1$	$n = 20$, $\Delta t = 0.0001$	4.29742E−04	2.31338E−03
	$t = 0.1$	$n = 40$, $\Delta t = 0.0001$	3.74392E−05	1.60769E−03
	$t = 0.1$	$n = 60$, $\Delta t = 0.0001$	3.44366E−05	1.56552E−03

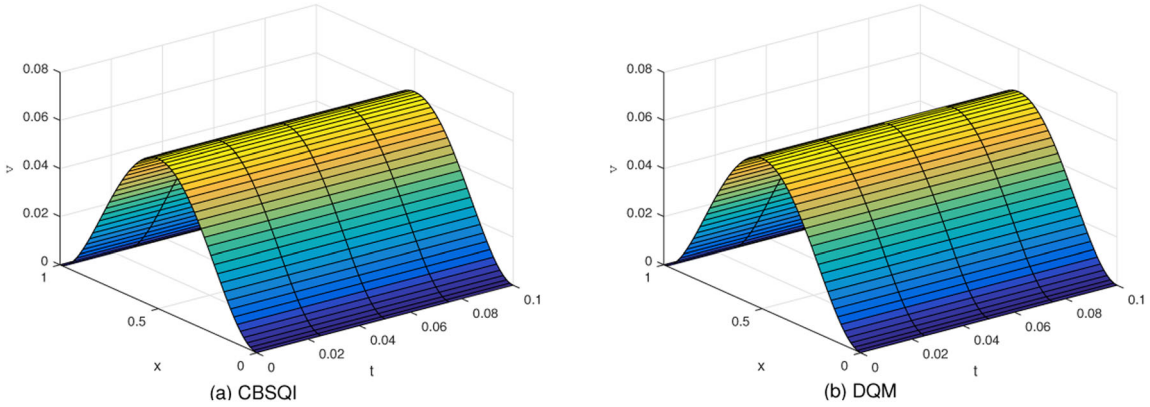

Fig. 6 Displacement for $t \leq 0.1$ ($n = 60$, $\Delta t = 0.0001$)

Table 6 Maximum absolute errors in $v(x, t)$ and $v_{xx}(x, t)$ for Problem 4

Methods	Time	Parameters	v	v_{xx}
Method 1	$t = 0.1$	$n = 20$, $\Delta t = 0.00001$	2.5279E−06	1.6270E−05
CBSQI	$t = 0.1$	$n = 40$, $\Delta t = 0.00001$	6.7106E−07	5.3848E−06
	$t = 0.1$	$n = 60$, $\Delta t = 0.00001$	3.0184E−07	2.5152E−06
	$t = 0.1$	$n = 90$, $\Delta t = 0.00001$	1.3481E−07	1.1447E−06
	$t = 0.1$	$n = 180$, $\Delta t = 0.00001$	3.3814E−08	2.9018E−07
	$t = 0.1$	$n = 270$, $\Delta t = 0.00001$	1.5037E−08	1.2931E−07
	$t = 1$	$n = 20$, $\Delta t = 0.00001$	1.67138E−03	1.51446E−02
	$t = 1$	$n = 40$, $\Delta t = 0.00001$	4.2926E−04	3.94205E−03
	$t = 1$	$n = 60$, $\Delta t = 0.00001$	1.9158E−04	1.76484E−03
	$t = 1$	$n = 90$, $\Delta t = 0.00001$	8.5292E−05	7.8691E−04
	$t = 1$	$n = 180$, $\Delta t = 0.00001$	2.1343E−05	1.9711E−04
	$t = 1$	$n = 270$, $\Delta t = 0.00001$	9.4875E−06	8.7634E−05
Method 2	$t = 0.1$	$n = 20$, $\Delta t = 0.00001$	2.0771E−06	6.24906E−05
DQM	$t = 0.1$	$n = 40$, $\Delta t = 0.00001$	6.09463E−07	1.51773E−05
	$t = 0.1$	$n = 60$, $\Delta t = 0.00001$	3.68338E−07	6.65529E−06
	$t = 1$	$n = 20$, $\Delta t = 0.00001$	8.95168E−04	6.65292E−03
	$t = 1$	$n = 40$, $\Delta t = 0.00001$	2.06595E−04	1.53744E−03
	$t = 1$	$n = 60$, $\Delta t = 0.00001$	9.13609E−05	6.89179E−04

$$\phi(x,0) = 0, \quad \phi(0,t) = \phi(1,t) = 0.$$
$$\psi(x,0) = 2 - 12x + 12x^2, \quad \psi(0,t) = \psi(1,t) = 2\cos t.$$

The computed results obtained by both methods and analytical solution are compared in Table 5.

Figure 6 depicts the computed numerical results for $n = 60$, $\Delta t = 0.0001$ at $t = 0.1$.

Problem 4 Consider a fourth-order nonhomogeneous PDE

$$\frac{\partial^2 v}{\partial t^2} + (1+x)\frac{\partial^4 v}{\partial x^4} = f(x,t), \quad x \in [0,1], \quad t > 0$$
where $f(x,t) = -72(1+x)(1-5x+5x^2)t\sin t + (x-x^2)^3(2\cos t - t\sin t)$.

The initial and boundary conditions are as follows

$$v(x,0) = 0, \quad v_t(x,0) = 0, \qquad x \in [0,1],$$
$$v(0,t) = v(1,t) = v_{xx}(0,t) = v_{xx}(1,t) = 0, \qquad t \geq 0,$$

The analytical solution is $v(x,t) = (x-x^2)^3 t\sin t$. From Eq. (4), initial and boundary conditions are derived as

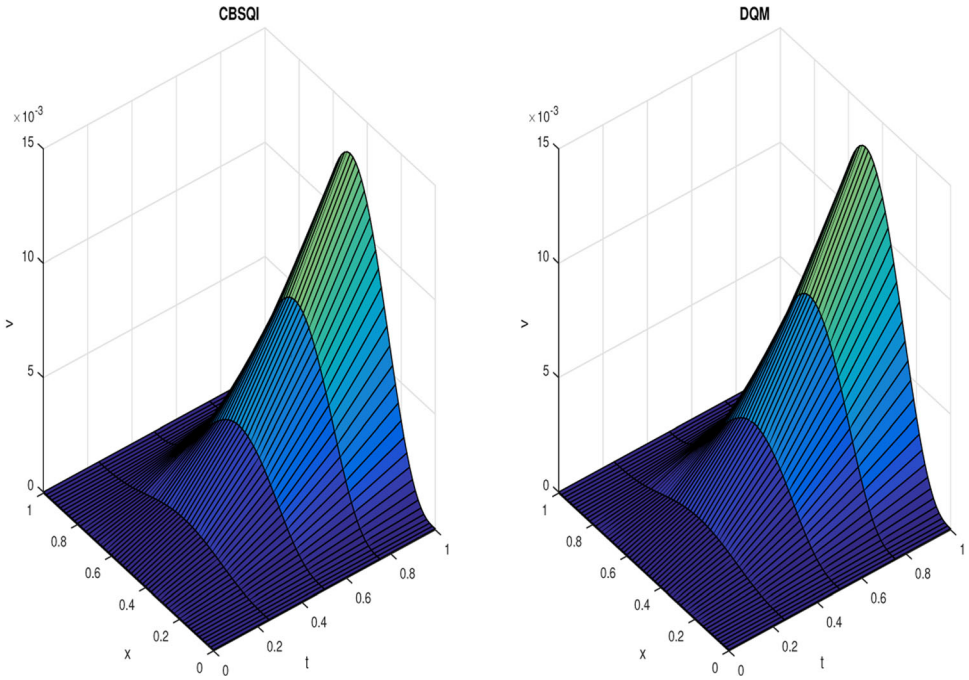

Fig. 7 Displacement for $t \leq 1.0$ ($n = 60$, $\Delta t = 0.0001$)

$$\phi(x,0) = 0, \quad \phi(0,t) = \phi(1,t) = 0.$$
$$\psi(x,0) = 0, \quad \psi(0,t) = \psi(1,t) = 0.$$

The computed results obtained by both methods and analytical solutions are compared in Table 6.

Figure 7 depicts the computed numerical results for $n = 60$, $\Delta t = 0.0001$ at $t = 1$.

Thus, according to the given tabular results and figures, we conclude that for each problem, DQM gives better solutions than CBSQI for the small number of grid points, but we also found that for a large number of grid points, DQM becomes unstable, whereas CBSQI produces good solutions.

9 Conclusions

In this paper, we have presented two numerical methods named CBSQI and DQM for solving fourth-order parabolic PDEs. The proposed methods are tested on four test problems, and on the basis of these results, we summarize the final outcomes as

1. DQM gives better solutions than CBSQI when the number of grid points is small, but we also found that for a large number of grid points, DQM becomes unstable, whereas CBSQI produces good solutions.
2. The stability of both the methods is discussed, and it is found that DQM is conditionally stable, whereas CBSQI is unconditionally stable.
3. To the best of the authors' knowledge, CBSQI Crank–Nicholson scheme technique has been used for the first time for solving fourth-order parabolic PDEs. The main advantage of CBSQI is that it is very easy to implement.

(iv) The proposed methods can be applied for higher dimensional problems.

Acknowledgements SK thanks Council of Scientific and Industrial Research (CSIR), Government of India [File No: 09/143(0889)/2017-EMR-I], for the financial support given during this work.

References

1. Collatz L (1973) Hermitian methods for initial value problems in partial differential equations topics in numerical analysis. Academic Press, London, pp 41–61
2. Conte SD (1957) A stable implicit finite difference approximation to a fourth order parabolic equation. J Assoc Comput Mech 4:18–23
3. Crandall SH (1954) Numerical treatment of a fourth order partial differential equations. J Assoc Comput Mech 1:111–118
4. Evans DJ (1965) A stable explicit method for the finite difference solution of a fourth order parabolic partial differential equation. Comput J 8:280–287
5. Todd J (1956) A direct approach to the problem of stability in the numerical solution of partial differential equations. Commun Pure Appl Math 9:597–612
6. Jain MK, Iyengar SRK, Lone AG (1976) Higher order difference formulas for a fourth order parabolic partial differential equation. Int J Numer Methods Eng 10:1357–1367
7. Fairweather G, Gourlay AR (1967) Some stable difference approximations to a fourth order parabolic partial differential equation. Math Comput 21:1–11

8. Lees M (1961) Alternate direction and semi explicit difference methods for solving parabolic partial differential equation. Numer Math 3:398–412
9. Mohanty RK, McKee S, Kaur D (2017) A class of two-level implicit unconditionally stable methods for a fourth order parabolic equation. Appl Math Comput 309:272–280
10. Mittal RC, Jain RK (2011) B-splines methods with redefined basis functions for solving fourth order parabolic partial differential equations. Appl Math Comput 217:9741–9755
11. Dehghan M, Manafian J (2009) The solution of the variable coefficients fourth-order parabolic partial differential equations by homotopy perturbation method. Zeitschrift fr Naturforschung A 64:420–430
12. Sablonniere P (2005) Univariate spline quasi-interpolants and applications to numerical analysis. Rend Semin Mat Univ Politec Torino 63:211–222
13. Sablonniere P (2007) A quadrature formula associated with a univariate spline quasi interpolant. BIT 47:825–837
14. Zhu CG, Kang WS (2010) Applying cubic B-spline quasi-interpolation to solve hyperbolic conservation laws. UPB Sci Bull Ser 72:49–58
15. Zhu CG, Kang WS (2010) Numerical solution of Burgers–Fisher equation by cubic B-spline quasi-interpolation. Appl Math Comput 216:2679–2686
16. Kumar R, Baskar S (2016) B-spline quasi-interpolation based numerical methods for some sobolev type equations. J Comput Appl Math 292:41–66
17. Bellman R, Kashef BG, Casti J (1972) Differential quadrature: a technique for the rapid solution of nonlinear differential equations. J Comput Phys 10:40–52
18. Quan JR, Chang CT (1989) New insights in solving distributed system equations by quadrature methods-I. Comput Chem Eng 13:779–788
19. Mittal RC, Jiwari R (2009) Numerical study of Fisher's equation by using differential quadrature method. Int J Inf Syst Sci 5:143–160
20. Mittal RC, Jiwari R (2012) A differential quadrature method for numerical solutions of Burgers'-type equations. Int J Numer Methods Heat Fluid Flow 22:880–895
21. Falco MD, Gaeta M, Loia V, Rarita L (2016) Differential quadrature based numerical solutions of a fluid dynamic model for supply chains. Commun Math Sci 14:1467–1476
22. Ghasemi M (2018) On the numerical solution of high order multi-dimensional elliptic PDEs. Comput Math Appl 76:1228–1245
23. Jiwari R, Tomasiello S, Tornabene F (2018) A numerical algorithm for computational modelling of coupled advection–diffusion–reaction systems. Eng Comput 35:1383–1401
24. Macias-Diaz JE, Tomasiello S (2016) A differential quadrature-based approach a la Picard for systems of partial differential equations associated to fuzzy differential equations. J Comput Appl Math 299:15–23
25. Dehghan M, Abbaszadeh M (2016) Variational multiscale element free Galerkin (VMEFG) and local discontinuous Galerkin (LDG) methods for solving two-dimensional Brusselator reaction–diffusion system with and without cross-diffusion. Comput Methods Appl Mech Eng 300:770–797
26. Dehghan M, Abbaszadeh M (2018) Solution of multi-dimensional Klein–Gordon–Zakharov and Schrodinger/Gross–Pitaevskii equations via local radial basis functions–differential quadrature (RBF-DQ) technique on non-rectangular computational domains. Eng Anal Bound Elem 92:156–170
27. Dehghan M, Mohammadi V (2015) The numerical solution of Cahn–Hilliard (CH) equation in one, two and three-dimensions via globally radial basis functions (GRBFs) and RBFs-differential quadrature (RBFs-DQ) methods. Eng Anal Bound Elem 51:74–100
28. Dehghan M, Nilpour A (2013) Numerical solution of the system of second-order boundary value problems using the local radial basis functions based differential quadrature collocation method. Appl Math Model 37:8578–8599
29. Lakestani M, Dehghan M (2009) Numerical solution of Fokker–Planck equation using the cubic B-spline scaling functions. Numer Methods Partial Differ Equ 25:418–429
30. Schumaker L (2007) Spline functions: basic theory. Cambridge University Press, Cambridge
31. Sablonniere P (2003) Quadratic spline quasi-interpolants on bounded domains of Rd, d = 1; 2; 3. Rend Semin Mat Univ Politec Torino 61:229–246
32. Shu C (2000) Differential quadrature and its application in engineering. Springer, London

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.