



ARTICLE

A Comparative Numerical Study of Parabolic Partial Integro-Differential Equation Arising from Convection-Diffusion

Kamil Khan¹, Arshed Ali^{1,*}, Fazal-i-Haq², Iltaf Hussain³ and Nudrat Amir⁴

¹Department of Mathematics, Islamia College, Peshawar, 25000, Pakistan

²Department of Mathematics, Statistics and Computer Science, The University of Agriculture, Peshawar, 25000, Pakistan

³Department of Basic Sciences & Islamiat, University of Engineering & Technology, Peshawar, 25000, Pakistan

⁴Department of Basic Sciences & Humanities, CECOS University of Information Technology and Emerging Sciences, Peshawar, 25000, Pakistan

*Corresponding Author: Arshed Ali. Email: arshad.ali@icp.edu.pk

Received: 10 July 2020 Accepted: 10 November 2020

ABSTRACT

This article studies the development of two numerical techniques for solving convection-diffusion type partial integro-differential equation (PIDE) with a weakly singular kernel. Cubic trigonometric B-spline (CTBS) functions are used for interpolation in both methods. The first method is CTBS based collocation method which reduces the PIDE to an algebraic tridiagonal system of linear equations. The other method is CTBS based differential quadrature method which converts the PIDE to a system of ODEs by computing spatial derivatives as weighted sum of function values. An efficient tridiagonal solver is used for the solution of the linear system obtained in the first method as well as for determination of weighting coefficients in the second method. An explicit scheme is employed as time integrator to solve the system of ODEs obtained in the second method. The methods are tested with three nonhomogeneous problems for their validation. Stability, computational efficiency and numerical convergence of the methods are analyzed. Comparison of errors in approximations produced by the present methods versus different values of discretization parameters and convection-diffusion coefficients are made. Convection and diffusion dominant cases are discussed in terms of Peclet number. The results are also compared with cubic B-spline collocation method.

KEYWORDS

Partial integro-differential equation; convection-diffusion; collocation method; differential quadrature; cubic trigonometric B-spline functions; weakly singular kernel

1 Introduction

Partial integro-differential equations have been applied to model different physical phenomena in science and engineering such as heat conduction [1], reactor dynamics [2], flow in fractured



biomaterials [3], electricity swaptions [4], financial option pricing [5], viscoelasticity [6], population dynamics [7], and convection-diffusion [8–11].

Convection-diffusion integro-differential equation has been used to model several important physical phenomena such as heat and mass transfer, flows in porous media, current density in fluids, and pollutant transport in atmosphere, streams, rivers and oceans (see [8,10,11] and the references therein).

PIDE models have been rarely solved analytically and their general solution is only obtained under restrictive conditions [12]. Due to such restrictive conditions the models become over simplified and also limit their physical relevance. Therefore, alternative approaches have been obtained for the solution of PIDEs including finite difference methods [13,14], finite element method [15], radial basis function collocation methods [16], spectral method [17], quasi-wavelet method [18], and spline methods [19–21].

This paper is devoted to comparative study of two efficient numerical techniques for solution of convection-diffusion parabolic PIDE [8,10] defined by:

$$\frac{\partial u(x, t)}{\partial t} + c \frac{\partial u(x, t)}{\partial x} - d \frac{\partial^2 u(x, t)}{\partial x^2} = \int_0^t K(q - t) u(x, q) dq + f(x, t), \quad x \in \Gamma = [a, b], \quad t > 0, \quad (1)$$

with initial condition

$$u(x, 0) = g(x), \quad x \in \Gamma, \quad (2)$$

boundary conditions

$$u(a, t) = g_1(t), \quad u(b, t) = g_2(t), \quad t > 0, \quad (3)$$

where the integral term is known as memory term, $g(x), g_1(t), g_2(t)$ are known functions, and $K(q, t) = (t - q)^{-\mu}$, $0 < \mu < 1$, is weakly singular kernel. The parameters c and d are positive constants which represent quantities of the convection and diffusion processes respectively. Different numerical techniques are developed to obtain the solution of the PIDE (1). Siddiqi et al. [8] employed cubic b-spline functions to spatial derivatives and Euler backward formula to time derivative to solve the PIDE (1). Ali et al. [9] constructed quartic B-spline collocation technique to obtain the solution of the PIDE (1). Fahim et al. [10] have solved (1) using product trapezoidal integration rule and sinc collocation method. Recently, Al-Humedi et al. [11] have solved (1) using cubic B-spline Galerkin method with quadratic weight function.

The cubic trigonometric B-spline methods are numerically accurate, computationally fast, consistent, and has ability to get the approximate solution at any point in the domain with more accuracy as compared to the conventional finite difference method which approximates solution at only selected points in the domain. In recent years solution of several partial differential equations have been found using these methods [22–28] due to special geometric features of trigonometric splines [22,29–31]. In 1964, Schoenberg [29] introduced piecewise trigonometric spline functions and founded the theory of locally supported trigonometric splines, named as trigonometric B-splines. It was derived that trigonometric spline can be written as linear combination of trigonometric B-splines. Lycche et al. [32] derived a recurrence relation using trigonometric B-splines divided differences for stable computation of trigonometric splines. They also formulated derivatives of trigonometric B-splines and provided an integral representation of trigonometric divided differences with trigonometric B-spline as kernel. Later on several other researchers have enhanced the theory of trigonometric B-splines by further developing many important properties

such as C^2 continuity, nonnegativity, partition of unity, smoothness, curve control and its analysis, and banded interpolation matrix (see [22,30,31] and the references therein). The first application of cubic trigonometric B-spline functions appeared in the literature in 2010 [33] when these functions were used for the solution of Two-point boundary value problem. In 2014, CTBS functions were extended for the solution of initial and boundary-value hyperbolic problems [22]. Recently, CBTS functions have been employed for the solution of PIDE problem [19,34]. The CTBS functions are also employed for the solution different problems which include Hunter Saxton equation [25], time fractional diffusion-wave equation [26], Fisher's reaction-diffusion equations [28], non-conservative linear transport problems [27], Nonclassical diffusion problems [35], Hyperbolic telegraph equation [36], Fisher's equations [37], and coupled Burgers' equations [38].

In 1937, Frazer et al. [39] initiated the use of collocation method. This method is a special variant of the method of weighted residuals which uses dirac delta functions as weighting functions to minimize the residual. The main advantage of the collocation method is that as the number of collocation points increases, more points from the domain satisfy the functional equation. As a result approximate solution obtained by this method approaches to exact solution. Another advantage of this method is that using the dirac delta function makes computation of residual easy. The method is coupled with other methods such as Galerkin method, least squares method, Newton's method, and finite difference method for the solution of various problems (see [22,40,41] and the references therein). In this paper, the first technique is developed by coupling CTBS functions with collocation procedure (CTBS-CT) for the solution of problem (1). The main advantage of this technique is that it reduces (1) to a tridiagonal system of linear equations which can be solved by an efficient tridiagonal solver.

The differential quadrature (DQ) method was projected as strong alternative of conventional methods i.e., finite difference and finite element methods. This method produces accurate numerical results with little computational effort as it uses smaller set of grid points while the conventional methods need relatively larger sets [42,43]. The DQ method was first introduced by Bellman et al. [44] in 1971, for approximating derivative of a sufficiently smooth function. The basic structure of DQ method is that it uses a weighted linear sum of functional values to approximate the function derivatives. The weighting coefficients play a key role since accuracy of the approximation depends on the accuracy of these coefficients. Initially Bellman and his co-researchers have developed two methods for determination of weighting coefficients [45]. However, the methods lead to ill-conditioned algebraic system for larger set of grid points. Shu [46] presented a systematic theoretical analysis and applications of earlier work about this method, and introduced two explicit approaches (i) a recurrence formula, and (ii) matrix multiplication, for computing the weighting coefficients of higher order derivatives to avoid the ill-conditioned system. The selection of grid points also effects accuracy of the DQ method and better accuracy was achieved using non-uniform and scattered grid points, and ghost points [34,43,47,48]. The DQ method is extended for the solution of various problems including Fisher's reaction-diffusion equations [28], coupled viscous Burgers' equations [38], space-dependent inverse heat problems [48], Kawahara equation [49], and shock wave simulations [50].

Recently, Korkmaz et al. [27] pioneered DQ method along with CTBS functions for the solution of second order advection-diffusion equation. Tasmir et al. [28] presented modified CTBS functions based DQ method for second order nonlinear Fisher's reaction-diffusion equations. Kumar Singh et al. [38] extended the method for the solution of one and two dimensional coupled viscous Burgers' equations. Arora et al. [51] developed a hybrid CTBS based DQ method for the solution of nonlinear Burgers' equations. In this paper, the other technique is developed by

coupling CTBS functions with differential quadrature (CTBS-DQ) approach. The key feature of this approach is that it reduces (1) to a system of ODEs which can be solved by any ODE solver.

Upto the best of our knowledge a comparative study of PIDE (1) using the proposed techniques has not been done. Moreover, solving PIDEs of the form (1) numerically have two major challenges in addition to discretization of temporal and spatial derivatives: the singular kernel and approximation of the memory term which also affect stability and accuracy of a numerical method.

Remaining work of the paper is arranged as follows: Section 2 shows development of the proposed methods through combination of CTBS functions with collocation and differential quadrature procedures. Section 3 analyzes stability of the present schemes. Section 4 shows numerical results of both methods including error norms, computational efficiency, conditioning, spectral radii, convergence and comparison with an existing method in order to establish the current approaches. Section 5 concludes the findings and outcomes of the paper.

2 The Proposed Methods

In order to develop, the proposed methods, consider the problem (1)–(3). The spatial domain Γ is divided into N sub-intervals $\Gamma_n = [x_{n-1}, x_n]$, $n = 1, 2, \dots, N$, of equal length $h = \frac{b-a}{N}$ by the nodes x_n , $n \in I = \{0, 1, \dots, N\}$ with $a = x_0$ and $b = x_N$. Let $t^l = l\delta t$, $l = 0, 1, 2, \dots, M$, where δt is temporal step.

2.1 CTBS-CT

Taking $t = t^{l+1}$ in Eq. (1) and the temporal derivative is approximated by Euler backward formula which reduces Eq. (1) to the following form:

$$\frac{u(x, t^{l+1}) - u(x, t^l)}{\delta t} + c \frac{\partial u(x, t^{l+1})}{\partial x} - d \frac{\partial^2 u(x, t^{l+1})}{\partial x^2} = \int_0^{t^{l+1}} (t^{l+1} - q)^{-\mu} u(x, q) dq + f(x, t^{l+1}). \quad (4)$$

The integral part of Eq. (4) is approximated as below ([9]):

$$\begin{aligned} \int_0^{t^{l+1}} (t^{l+1} - q)^{-\mu} u(x, q) dq &= \sum_{s=0}^l \int_{t^s}^{t^{s+1}} q^{-\mu} u(x, t^{l+1} - q) dq, \\ &\approx \sum_{s=0}^l u(x, t^{l-s+1}) \int_{t^s}^{t^{s+1}} q^{-\mu} dq, \\ &\approx \sum_{s=0}^l b_s u^{l-s+1}, \end{aligned} \quad (5)$$

where u^l denotes $u(x, t^{l+1})$ and $b_s = \frac{(\delta t)^{1-\mu}}{1-\mu} \left((s+1)^{1-\mu} - s^{1-\mu} \right)$ and $b_s \rightarrow 0$ as $s \rightarrow \infty$.

Putting Eq. (5) in Eq. (4), we get

$$\frac{u^{l+1} - u^l}{\delta t} + c \frac{\partial u^{l+1}}{\partial x} - d \frac{\partial^2 u^{l+1}}{\partial x^2} = \sum_{s=0}^l b_s u^{l-s+1} + f^{l+1}, \quad (6)$$

where $f^{l+1} = f(x, t^{l+1})$.

Simplifying Eq. (6), we get

$$u^{l+1} + \delta t \left(cu_x^{l+1} - du_{xx}^{l+1} - b_0 u^{l+1} \right) = (1 + b_1 \delta t) u^l + \delta t \left(\sum_{s=2}^l b_s u^{l-s+1} + f^{l+1} \right), \quad (7)$$

where $u_x^{l+1} = \frac{\partial u^{l+1}}{\partial x}$ and $u_{xx}^{l+1} = \frac{\partial^2 u^{l+1}}{\partial x^2}$.

Now collocating Eq. (7) at $x = x_r$, $r \in I$, we obtain

$$(1 - b_0 \delta t) u_r^{l+1} + \delta t \left(c(u_x)_r^{l+1} - d(u_{xx})_r^{l+1} \right) = (1 + b_1 \delta t) u_r^l + \delta t \left(\sum_{s=2}^l b_s u_r^{l-s+1} + f_r^{l+1} \right), \quad (8)$$

where $u_r^{l+1} = u(x_r, t^{l+1})$. Next using CTBS functions $S_r(x)$, given in [22,23] to approximate $u(x_r, t)$.

Let

$$u(x, t) = \sum_{r=-1}^{N+1} C_r(t) S_r(x), \quad (9)$$

where

$$S_r(x) = \frac{1}{\omega} \begin{cases} \phi_{r-2}^3, & x \in \Gamma_{r-1}, \\ \phi_{r-2}[\phi_{r-2}\rho_r + \rho_{r+1}\phi_{r-1}] + \rho_{r+1}\phi_{r-1}^2, & x \in \Gamma_r, \\ \phi_{r-2}\rho_{r+1}^2 + \rho_{r+2}[\phi_{r-1}\rho_{r+1} + \rho_{r+2}\phi_r], & x \in \Gamma_{r+1}, \\ \rho_{r+2}^3, & x \in \Gamma_{r+2}, \\ 0, & \text{otherwise,} \end{cases}$$

$$\phi_r = \sin\left(\frac{x - x_r}{2}\right), \quad \rho_r = \sin\left(\frac{x_r - x}{2}\right), \quad \text{and} \quad \omega = \sin\left(\frac{h}{2}\right) \sin(h) \sin\left(\frac{3h}{2}\right).$$

Lemma [22,33]: The values of $S_r(x)$, $S'_r(x)$ and $S''_r(x)$ at the node x_s are obtained as:

$$S_r(x_s) = \begin{cases} \delta_2, & \text{if } r - s = 0, \\ \delta_1, & \text{if } r - s = 1 \text{ or } -1, \\ 0, & \text{elsewhere,} \end{cases}$$

$$S'_r(x_s) = \begin{cases} \delta_4, & \text{if } r - s = 1, \\ \delta_3, & \text{if } r - s = -1, \\ 0, & \text{elsewhere,} \end{cases}$$

and

$$S_r''(x_s) = \begin{cases} \delta_6, & \text{if } r-s=0, \\ \delta_5, & \text{if } r-s=1 \text{ or } -1, \\ 0, & \text{elsewhere,} \end{cases}$$

Then using Eq. (9), the function u and its derivatives at the r th node ($x = x_r$, $r \in I$) can be expressed as

$$\begin{aligned} u_r &= \delta_1 C_{r-1} + \delta_2 C_r + \delta_1 C_{r+1}, \\ u_r' &= \delta_3 C_{r-1} + \delta_4 C_{r+1}, \\ u_r'' &= \delta_5 C_{r-1} + \delta_6 C_r + \delta_5 C_{r+1}, \end{aligned} \quad (10)$$

$$\begin{aligned} \text{where } \delta_1 &= \frac{\sin^2(\frac{h}{2})}{\sin(h) \sin(\frac{3h}{2})}, \quad \delta_2 = \frac{2}{1+2\cos(h)}, \quad \delta_3 = -\frac{3}{4\sin(\frac{3h}{2})}, \quad \delta_4 = \frac{3}{4\sin(\frac{3h}{2})}, \\ \delta_5 &= \frac{3(1+3\cos(h))}{16(2\cos(\frac{h}{2}) + \cos(\frac{3h}{2})) \sin^2(\frac{h}{2})}, \quad \text{and } \delta_6 = -\frac{3\cot^2(\frac{h}{2})}{2+4\cos(h)}. \end{aligned}$$

Thus by using (10) in (8) and combining like terms, we get

$$\alpha_1 C_{r-1}^{l+1} + \alpha_2 C_r^{l+1} + \alpha_3 C_{r+1}^{l+1} = (1+b_1\delta t)(\delta_1 C_{r-1}^l + \delta_2 C_r^l + \delta_1 C_{r+1}^l) + F_r, \quad r \in I, \quad (11)$$

where $\alpha_1 = [(1-b_0\delta t)\delta_1 + c\delta t\delta_3 - d\delta t\delta_5]$, $\alpha_2 = [(1-b_0\delta t)\delta_2 - d\delta t\delta_6]$, $\alpha_3 = [(1-b_0\delta t)\delta_1 + c\delta t\delta_4 - d\delta t\delta_5]$, and $F_r = \delta t \sum_{s=2}^l b_s [\delta_1 C_{r-1}^{l-s+1} + \delta_2 C_r^{l-s+1} + \delta_1 C_{r+1}^{l-s+1}] + \delta t f_r^{l+1}$.

For $r=0$ and N , Eq. (11) reduces to $\alpha_1 C_{-1}^{l+1} + \alpha_2 C_0^{l+1} + \alpha_3 C_1^{l+1} = F_0$ and $\alpha_1 C_{N-1}^{l+1} + \alpha_2 C_N^{l+1} + \alpha_3 C_{N+1}^{l+1} = F_N$ respectively. After eliminating the parameters C_{-1}^{l+1} and C_{N+1}^{l+1} using (10) and the boundary conditions (3), we get the linear system of $N+1$ equations in $N+1$ unknowns whose matrix form is as follows:

$$\mathbf{AC}^{l+1} = \mathbf{BC}^l + \mathbf{F}, \quad l \geq 0, \quad (12)$$

where

$$\mathbf{A} = \begin{bmatrix} \alpha_2 - \frac{\delta_2}{\delta_1} \alpha_1 & \alpha_3 - \alpha_1 & 0 & 0 & \dots & 0 & 0 & 0 \\ \alpha_1 & \alpha_2 & \alpha_3 & 0 & \dots & 0 & 0 & 0 \\ 0 & \alpha_1 & \alpha_2 & \alpha_3 & \dots & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \dots & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \dots & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \dots & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & \dots & \alpha_1 & \alpha_2 & \alpha_3 \\ 0 & 0 & 0 & 0 & \dots & 0 & \alpha_1 - \alpha_3 & \alpha_2 - \frac{\delta_2}{\delta_1} \alpha_3 \end{bmatrix},$$

$$\mathbf{B} = (1 + b_1 \delta t) \begin{bmatrix} 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \delta_1 & \delta_2 & \delta_1 & 0 & \dots & 0 & 0 & 0 \\ 0 & \delta_1 & \delta_2 & \delta_1 & \dots & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \dots & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \dots & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \dots & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & \dots & \delta_1 & \delta_2 & \delta_1 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{C}^{l+1} = [C_0^{l+1}, C_1^{l+1}, \dots, C_N^{l+1}]^T \text{ and } \mathbf{F} = [F_0 + (1 + b_1 \delta t)g_1(t^l) - \frac{\alpha_1}{\delta_1}g_1(t^{l+1}), F_1, F_2, \dots, F_{N-1}, F_N + (1 + b_1 \delta t)g_2(t^l) - \frac{\alpha_3}{\delta_1}g_2(t^{l+1})]^T.$$

Eq. (12) can be re-written as

$$\mathbf{C}^{l+1} = \mathbf{D}\mathbf{C}^l + \mathbf{K}, \quad l \geq 0, \quad (13)$$

where $\mathbf{D} = \mathbf{A}^{-1}\mathbf{B}$ and $\mathbf{K} = \mathbf{A}^{-1}\mathbf{F}$. The system Eq. (12) is solved using the efficient tridiagonal solver “Thomas Algorithm.”

2.2 CTBS-DQ

The DQ approach provides k th order derivative of the function $u(x, t)$ from the values of $u(x, t)$ at $x_r, r \in I_1 = \{1, 2, \dots, N\}$, as

$$\frac{\partial^k u(x_r, t)}{\partial x^k} = \sum_{s=1}^N a_{rs}^{(k)} u(x_s, t), \quad (14)$$

where $a_{rs}^{(k)}$, $k = 1, 2, \dots$ are k th order weighting coefficients (WCs) which are determined by test functions. Different basis functions have been employed as test functions for determination of the WCs such as CTBS functions [27,38], polynomials [42], radial basis functions [48], cubic B-spline functions [49], and sinc functions [50]. In CTBS-DQ method we take the following test functions which are defined as modified CTBS functions [51]:

$$\varpi_1(x) = (S_1 + 2S_0)(x),$$

$$\varpi_2(x) = (S_2 - S_0)(x),$$

$$\varpi_l(x) = S_l(x), \quad \text{for } l = 3, 4, \dots, N-2,$$

$$\varpi_{N-1}(x) = (S_{N-1} - S_{N+1})(x),$$

$$\varpi_N(x) = (S_N + 2S_{N+1})(x),$$

which form a basis in region Γ .

Thus for each basis function $\varpi_l(x_r)$, Eq. (14) gives

$$\frac{\partial^k \varpi_m}{\partial x^k}(x_r) = \sum_{s=1}^N a_{rs}^{(k)} \varpi_m(x_s), \quad \text{for } r, m \in I_1, \text{ and } k = 1, 2,$$

which leads to the following matrix form:

$$G\vec{a}_r^{(k)} = H_r^{(k)}, \quad (15)$$

where $\vec{a}_r^{(k)} = [a_{rs}^{(k)} : s \in I_1]$, $H_r^{(k)} = \left[\frac{\partial^k \varpi_l}{\partial x^k}(x_r) : l \in I_1 \right]^T$, $G = [g_{rs} : r, s \in I_1]$ and

$$g_{rs} = \begin{cases} 2\delta_1 + \delta_2, & \text{if } r = s = 1 \text{ or } N, \\ \delta_2, & \text{if } r = s \text{ and } 1 < r, s < N, \\ \delta_1, & \text{if } (s = r - 1 \text{ and } 2 < r \leq N) \text{ or } (r = s - 1 \text{ and } 1 \leq s < N - 1), \\ 0, & \text{elsewhere.} \end{cases}$$

The weighting coefficients $a_{rs}^{(k)}$, $r, s \in I_1$, are obtained by solving the system (15) through the well known efficient tridiagonal solver “Thomas algorithm.” Now using (14) in (1) and taking $x = x_r$ to obtain the differential quadrature based technique,

$$\frac{\partial u(x_r, t)}{\partial t} + c \sum_{s=1}^N a_{rs}^{(1)} u(x_s, t) - d \sum_{s=1}^N a_{rs}^{(2)} u(x_s, \tau) = \int_0^t (t - q)^{-\mu} u(x_r, q) dq + f(x_r, t). \quad (16)$$

The explicit time marching process for Eq. (16) is obtained as follows:

$$\frac{u_r^{l+1} - u_r^l}{\delta t} = -c \vec{a}_r^{(1)} \mathbf{u}^l + d \vec{a}_r^{(2)} \mathbf{u}^l + \int_0^{t^{l+1}} (t^{l+1} - q)^{-\mu} u(x_r, q) dq + f_r^{l+1}, \quad (17)$$

where $\mathbf{u}^l = [u_r^l : r \in I_1]^T$. Using Eq. (5) in Eq. (17) and combining the like terms, we get

$$u_r^{l+1} = \frac{1}{1 - b_0 \delta t} \left(u_r^l + \delta t \left(-c \vec{a}_r^{(1)} \mathbf{u}^l + d \vec{a}_r^{(2)} \mathbf{u}^l + \sum_{s=1}^l b_s u_r^{l-s+1} + f_r^{l+1} \right) \right). \quad (18)$$

Eq. (18) leads to the following matrix form:

$$\mathbf{u}^{l+1} = \mathbf{D} \mathbf{u}^l + \mathbf{M}, \quad l \geq 0, \quad (19)$$

where

$$\mathbf{D} = \frac{1}{1 - b_0 \delta t} \begin{bmatrix} 1 + \delta t(b_1 - ca_{11}^{(1)} + da_{11}^{(2)}) & \delta t(-ca_{12}^{(1)} + da_{12}^{(2)}) & \dots & \delta t(-ca_{1N}^{(1)} + da_{1N}^{(2)}) \\ \delta t(-ca_{21}^{(1)} + da_{22}^{(2)}) & 1 + \delta t(b_1 - ca_{22}^{(1)} + da_{22}^{(2)}) & \dots & \delta t(-ca_{2N}^{(1)} + da_{2N}^{(2)}) \\ \vdots & \vdots & \dots & \vdots \\ \delta t(-ca_{N1}^{(1)} + da_{N1}^{(2)}) & \delta t(-ca_{N2}^{(1)} + da_{N2}^{(2)}) & \dots & 1 + \delta t(b_1 - ca_{NN}^{(1)} + da_{NN}^{(2)}) \end{bmatrix}$$

and $\mathbf{M} = [p_1, p_2, \dots, p_N]^T$, $p_r = \delta t(\sum_{s=2}^l b_s u_r^{l-s+1} + f_r^{l+1})$, $r \in I_1$.

3 Stability Analysis

In this section we assess stability of the present schemes (13) and (19). Let e^n , u^n and \tilde{u}^n represent error, exact solution and approximate solution at n th time level respectively, then $e^n = u^n - \tilde{u}^n$. The error equation for the methods (13) and (19) is given by

$$e^{n+1} = \mathbf{D}e^n, \quad (20)$$

where \mathbf{D} is the amplification matrix. The schemes (13) and (19) will be stable if $\|\mathbf{D}\| \leq 1$ (see [52] and Lax-Richtmyer condition for stability). This condition is equivalent to $\rho(\mathbf{D}) \leq 1$, where $\rho(\mathbf{D})$ represents spectral radius of the matrix \mathbf{D} [53] and is defined as $\rho(\mathbf{D}) = \max_{1 \leq i \leq N} |\lambda_i|$ and λ_i is an eigenvalue of \mathbf{D} . The values of λ_i depend on N , δt , c , d and μ . Computational values of $\rho(\mathbf{D})$ for different values of the parameters N , δt , c , d and μ are provided in the next section which show that the schemes (13) and (19) satisfy the condition of stability.

4 Test Problems

In this section three examples are taken from the literature [8] to validate the present approaches (13) and (19) for the solution of the problem (1)–(3) and comparison is also made with results obtained by cubic B-spline collocation method (CBSC) [8]. The methods are examined via E_∞ , E_2 error norms [8]. The schemes are also tested for convection and diffusion dominated problems via the Peclet number $P = \frac{c}{d}h$. High values of P lead to convection dominated problem while lower values of P lead to diffusion dominated problem [8]. Computations are carried out with Corei3, 2.4 GHz processor, 4 GB Ram, Matlab7.5 and computer run time (RT) is provided in seconds.

4.1 Problem 1

We take Eqs. (1)–(3) with convection-diffusion coefficients $c = 0.1$, $d = 0.1$ and $\mu = \frac{1}{4}$ elsewhere mentioned, and choose $f(x, t)$ such that the exact solution is given by [8]

$$u(x, t) = 2(t + 1) \sin^2 x, \quad 0 \leq x \leq 4\pi. \quad (21)$$

Table 1: Error norms for different values of convection and diffusion coefficients using $\delta t = 10^{-4}$, $M = 100$, $N = 100$

c	d	CTBS-DQ			CTBS-CT		
		E_∞	E_2	$\rho(\mathbf{D})$	E_∞	E_2	$\rho(\mathbf{D})$
0.5	0.1	$1.098e-03$	$1.758e-05$	1.0	$1.935e-05$	$1.125e-06$	1.0
0.2	0.1	$1.015e-03$	$1.445e-05$	1.0	$1.934e-05$	$1.125e-06$	1.0
0.1	0.1	$9.866e-04$	$1.395e-05$	1.0	$1.934e-05$	$1.125e-06$	1.0
0.05	0.1	$9.721e-04$	$1.382e-05$	1.0	$1.934e-05$	$1.125e-06$	1.0
0.01	0.1	$9.604e-04$	$1.378e-05$	1.0	$1.933e-05$	$1.125e-06$	1.0
0.1	0.5	$3.229e-03$	$4.590e-05$	1.0	$5.641e-05$	$3.485e-06$	1.0
0.1	0.2	$1.728e-03$	$2.442e-05$	1.0	$2.868e-05$	$1.704e-06$	1.0
0.1	0.05	$5.435e-04$	$7.704e-05$	1.0	$1.466e-05$	$8.524e-06$	1.0
0.1	0.01	$1.481e-04$	$2.665e-06$	1.0	$1.092e-05$	$6.569e-07$	1.0

Table 2: Error norms vs. N using $\delta t = 10^{-4}$, $M = 100$

N	CTBS-DQ			CTBS-CT		
	E_∞	E_2	RT (sec)	E_∞	E_2	RT (sec)
10	$2.257e-03$	$4.357e-04$	0.041	$1.154e-03$	$2.060e-04$	0.034
50	$1.161e-03$	$2.394e-05$	0.060	$4.776e-05$	$4.153e-06$	0.053
100	$9.866e-04$	$1.395e-05$	0.089	$1.934e-05$	$1.125e-06$	0.075
200	$6.974e-04$	$4.950e-06$	0.166	$1.236e-05$	$5.131e-07$	0.122
500	$2.662e-04$	$8.096e-07$	0.662	$1.038e-05$	$2.817e-07$	0.263

Table 3: Error norms at various time levels M using $\delta t = 10^{-4}$ and $N = 100$

M	CTBS-DQ			CTBS-CT		
	E_∞	E_2	RT (sec)	E_∞	E_2	RT (sec)
10	$1.109e-04$	$1.582e-06$	0.022	$1.406e-06$	$8.399e-08$	0.017
50	$5.256e-04$	$7.458e-06$	0.058	$8.940e-06$	$5.221e-07$	0.025
100	$9.866e-04$	$1.395e-05$	0.089	$1.934e-05$	$1.125e-06$	0.075
200	$1.756e-03$	$2.483e-05$	0.187	$4.135e-05$	$2.400e-06$	0.185
500	$3.300e-03$	$4.836e-05$	0.856	$1.118e-04$	$6.478e-06$	0.328

Table 4: Convergence in space using $\delta t = 10^{-5}$ and $M = 100$

N	CTBS-DQ			CTBS-CT			
	E_2	Order	$\rho(\mathbf{D})$	E_2	Order	$\kappa(\mathbf{A})$	$\rho(\mathbf{D})$
10	$4.342e-05$	—	1.0	$2.042e-05$	—	$8.964e+05$	1.0
50	$3.386e-06$	1.585	1.0	$3.620e-07$	2.506	$1.201e+04$	1.0
100	$1.580e-06$	1.100	1.0	$6.942e-08$	2.382	$2.848e+03$	1.0
200	$7.430e-07$	1.088	1.0	$1.701e-08$	2.029	$6.962e+02$	1.0
500	$2.261e-07$	1.298	1.0	$5.648e-09$	1.203	$1.100e+02$	1.0
1000	$6.588e-08$	1.780	1.0	$3.563e-09$	0.665	$2.729e+01$	1.0

Table 5: Convergence in time using $N = 100$ and $t = 0.1$

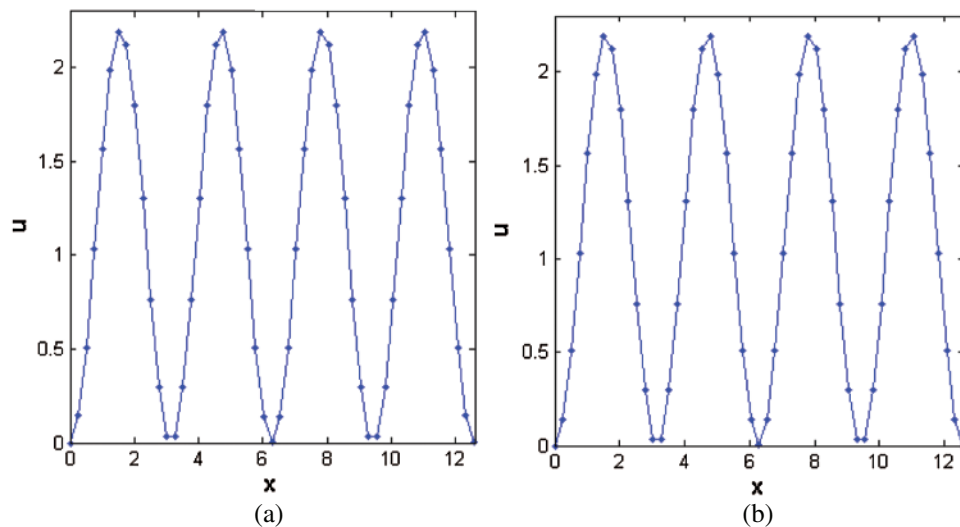
δt	CTBS-DQ			CTBS-CT			
	E_2	Order	$\rho(\mathbf{D})$	E_2	Order	$\kappa(\mathbf{A})$	$\rho(\mathbf{D})$
10^{-2}	$7.867e-05$	—	1.0	$1.033e-04$	—	$2.783e+00$	1.0
10^{-3}	$7.564e-05$	0.0171	1.0	$4.100e-05$	0.401	$2.836e+01$	1.0
10^{-4}	$7.609e-05$	-0.0026	1.0	$1.377e-05$	0.474	$2.847e+02$	1.0
10^{-5}	$7.614e-05$	-0.0003	1.0	$7.615e-06$	0.257	$2.848e+03$	1.0

Table 6: Error norms vs. the parameter μ using $\delta t = 10^{-4}$, $N = 100$ and $M = 100$

μ	CTBS-DQ			CTBS-CT			
	E_∞	E_2	$\rho(\mathbf{D})$	E_∞	E_2	$\kappa(\mathbf{A})$	$\rho(\mathbf{D})$
$\frac{1}{5}$	$9.847e-04$	$1.571e-05$	1.0	$9.458e-06$	$7.532e-07$	$2.847e+02$	1.0
$\frac{1}{4}$	$9.866e-04$	$1.395e-05$	1.0	$1.934e-05$	$1.125e-06$	$2.847e+02$	1.0
$\frac{1}{3}$	$9.911e-04$	$2.139e-05$	1.0	$1.176e-04$	$7.085e-06$	$2.847e+02$	1.0
$\frac{1}{2}$	$1.429e-03$	$8.913e-05$	1.0	$8.573e-04$	$5.256e-05$	$2.847e+02$	1.0

Table 7: Comparison of error norms with CBSC method [8], $\delta t = 10^{-5}$

P	N	M	CTBS-DQ		CTBS-CT		CBSC	
			E_∞	E_2	E_∞	E_2	E_∞	E_2
1.25663	10	10	$2.243e-05$	$4.340e-06$	$1.143e-05$	$2.040e-06$	$1.906e-05$	$3.785e-06$
		50	$1.122e-04$	$2.170e-05$	$5.717e-05$	$1.021e-05$	$9.328e-05$	$1.867e-05$
		100	$2.244e-04$	$4.342e-05$	$1.144e-04$	$2.042e-05$	$1.924e-04$	$3.680e-05$
0.25132	50	10	$1.197e-05$	$3.396e-07$	$3.662e-07$	$3.439e-08$	$9.466e-07$	$7.297e-08$
		50	$5.980e-05$	$1.696e-06$	$1.849e-06$	$1.738e-07$	$4.227e-06$	$2.892e-07$
		100	$1.194e-04$	$3.386e-06$	$3.719e-06$	$3.492e-07$	$7.136e-06$	$6.352e-07$

**Figure 1:** Numerical solutions obtained by (a) CTBS-DQ (b) CTBS-CT, using $N = 50$, $\delta t = 10^{-4}$, $M = 1000$, $c = 0.1$, $d = 0.1$, $\mu = 1/4$

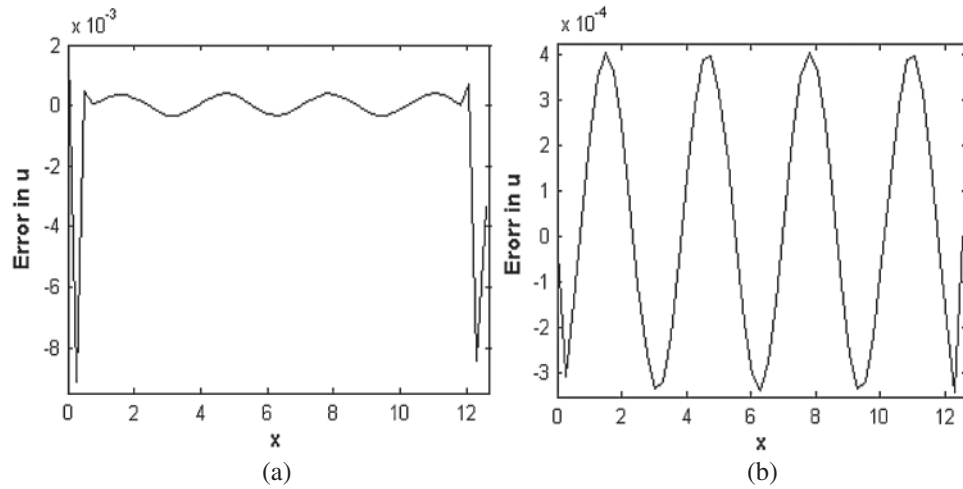


Figure 2: Error in numerical solutions obtained by (a) CTBS-DQ (b) CTBS-CT, using $N = 50$, $\delta t = 10^{-4}$, $M = 1000$, $c = 0.1$, $d = 0.1$, $\mu = 1/4$

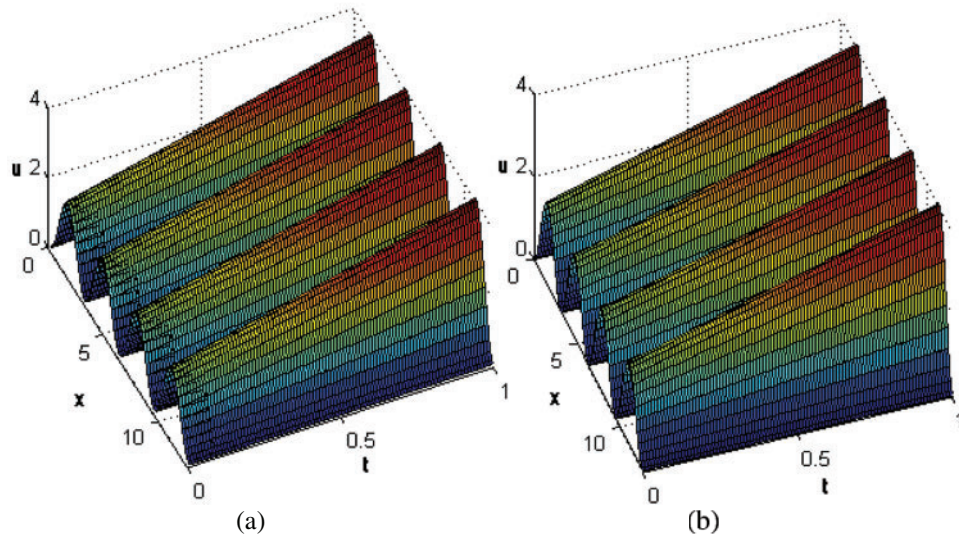


Figure 3: Numerical solutions obtained by (a) CTBS-DQ (b) CTBS-CT, over the time interval $[0, 1]$ using $N = 50$, $\delta t = 10^{-4}$, $c = 0.1$, $d = 0.1$, $\mu = 1/4$

The conditions g , g_1 , g_2 are found from Eq. (21). The error norms E_∞ , E_2 are obtained using the present methods (13) and (19) along with different values of the parameters number of nodes N , time step δt , time level M , convection and diffusion coefficients c , d , and μ which are reported in Tabs. 1–6. The effect of the convection and diffusion coefficient parameters c , d on the accuracy of both the methods are shown in Tab. 1. Computational efficiency RT of the methods are given in Tabs. 2 and 3 which shows that both the proposed methods are highly efficient. Numerical rate of convergence with respect to N and δt are provided in Tabs. 4 and 5, and it can be seen that the CTBS-CT has higher rate of convergence than the CTBS-DQ. However, the rate of convergence of CTBS-DQ becomes higher than that of CTBS-CT for larger values of spatial

nodes N . From Tabs. 4–6, it can be noted that numerical values of $\rho(\mathbf{D})$ for both methods satisfy the stability condition given in Section 3, whereas $\kappa(\mathbf{A})$ regarding CTBS-CT remains reasonably small. From Tab. 6, it can be noted that as the parameter μ decreases accuracy of the CTBS-DQ slightly increases as compared to the CTBS-CT. Furthermore, better accuracy of CTBS-CT than CBSC method [8] is evident from Tab. 7. The solution of (1) for cases of convection dominant and diffusion dominant problems are also provided in this table. It can also be seen from these tables that the CTBS-CT gives better results than the CTBS-DQ. Fig. 1 represents solutions obtained using CTBS-DQ and CTBS-CT at $M = 1000$. Fig. 2 displays errors in approximation obtained by the present methods. Fig. 3 shows the CTBS based solutions at different time levels. Fig. 4 represents condition numbers of system matrices of the methods. Fig. 5 compares convergence in space for the proposed methods.

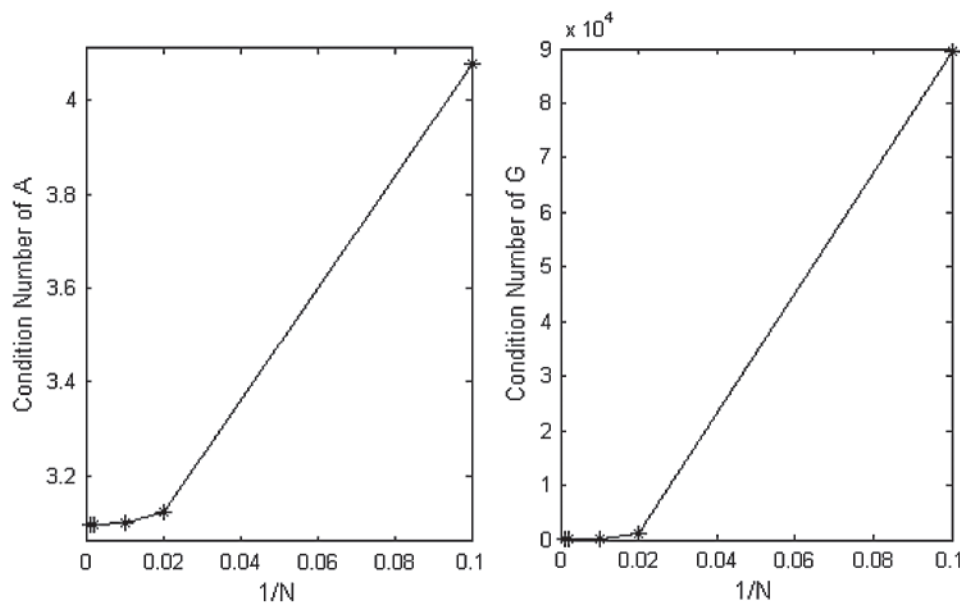


Figure 4: Condition numbers of system matrix A and interpolation matrix G corresponding to the methods (12) and (18) respectively, vs. the number of grid points N , for $\delta t = 10^{-4}$, $c = 0.1$, $d = 0.1$, $\mu = 1/4$

4.2 Problem 2

We take the Eqs. (1)–(3) with the convection-diffusion coefficients $c = 0.5$, $d = 0.005$, and $\mu = 1/3$. $f(x, t)$ is chosen so that the exact solution of (1)–(3) is given by [8]

$$u(x, t) = (t + 1) \cos \pi x, \quad 0 \leq x \leq 1, \quad t > 0. \quad (22)$$

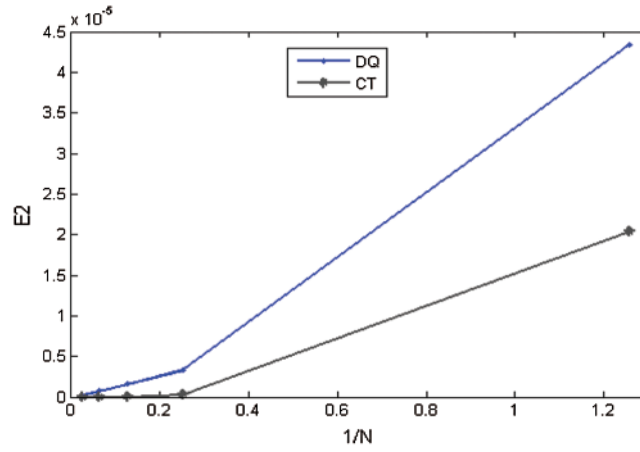


Figure 5: Convergence of the approximate solution in space using $\delta t = 10^{-4}$, $c = 0.1$, $d = 0.1$, $\mu = 1/4$

Table 8: Error norms using $M = 100$, $\mu = 1/3$

δt	N	CTBS-DQ		CTBS-CT	
		E_∞	E_2	E_∞	E_2
10^{-4}	10	$2.700e-03$	$2.951e-04$	$1.878e-05$	$3.443e-06$
	50	$4.973e-04$	$1.154e-05$	$1.512e-05$	$1.237e-06$
	100	$1.535e-04$	$2.471e-06$	$1.267e-05$	$8.654e-07$
	200	$4.117e-05$	$6.066e-07$	$1.234e-05$	$6.098e-07$
	500	$7.882e-06$	$2.538e-07$	$1.232e-05$	$3.853e-07$
10^{-5}	10	$2.615e-04$	$2.992e-05$	$7.158e-07$	$1.284e-07$
	50	$9.437e-05$	$1.968e-06$	$3.492e-07$	$2.770e-08$
	100	$5.734e-05$	$6.446e-07$	$3.286e-07$	$1.892e-08$
	200	$2.537e-05$	$1.660e-07$	$2.935e-07$	$1.325e-08$
	500	$5.595e-06$	$2.122e-08$	$2.653e-07$	$8.350e-09$

Table 9: Comparison of error norms with CBSC method [8] for $N = 100$, $P = 1$

δt	M	CTBS-DQ		CTBS-CT		CBSC	
		E_∞	E_2	E_∞	E_2	E_∞	E_2
10^{-4}	10	$1.565e-05$	$1.819e-07$	$7.694e-07$	$4.421e-08$	$1.171e-06$	$1.469e-08$
	50	$5.756e-05$	$6.492e-07$	$6.004e-06$	$3.789e-07$	$2.010e-05$	$2.543e-07$
	100	$8.706e-05$	$1.006e-06$	$1.267e-05$	$8.654e-07$	$7.266e-05$	$9.210e-07$
	500	$1.535e-04$	$2.471e-06$	$7.627e-05$	$5.256e-06$	$1.500e-03$	$1.906e-05$
10^{-5}	10	$1.640e-06$	$1.998e-08$	$1.734e-08$	$9.706e-10$	$3.013e-05$	$3.755e-07$
	50	$8.132e-06$	$9.519e-08$	$1.461e-07$	$8.282e-09$	$3.305e-04$	$5.006e-06$
	100	$1.554e-05$	$1.780e-07$	$3.286e-07$	$1.892e-08$	$1.061e-03$	$1.684e-05$
	500	$5.744e-05$	$6.446e-07$	$1.801e-06$	$1.144e-07$	$2.171e-02$	$2.841e-04$

The conditions g , g_1 , g_2 are found from Eq. (22). The error norms for different values of N , δt and M are computed which are reported in Tabs. 8, 9 along with the results of CBSC method [8]. It can be noted from Tab. 8 that errors produced by both the proposed methods decrease as number of nodes N increases. Also both the methods provide considerably better accuracy than the method [8]. Fig. 6 represents solutions obtained using CTBS-DQ and CTBS-CT at $M = 1000$. Fig. 7 depicts the solution at different time levels.

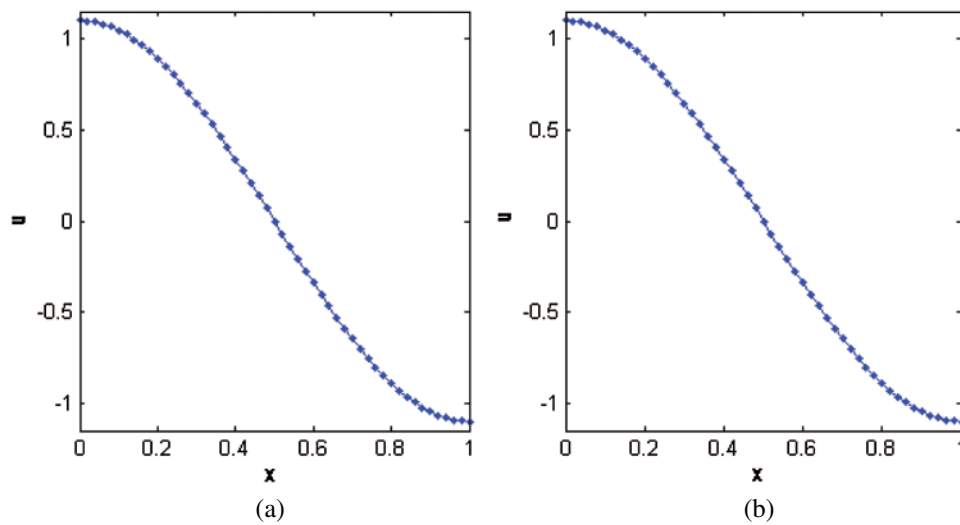


Figure 6: Numerical solutions obtained by (a) CTBS-DQ (b) CTBS-CT, using $N = 50$, $\delta t = 10^{-4}$, $M = 1000$, $c = 0.5$, $d = 0.005$, $\mu = 1/3$

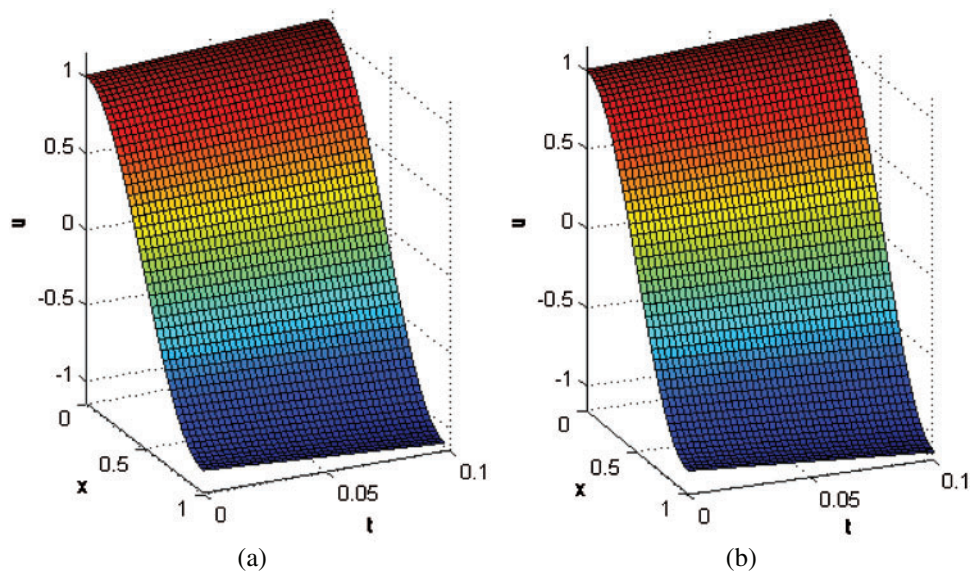


Figure 7: Numerical solutions obtained by (a) CTBS-DQ (b) CTBS-CT, over time interval $[0, 0.1]$ using $N = 50$, $\delta t = 10^{-4}$, $c = 0.5$, $d = 0.005$, $\mu = 1/3$

4.3 Problem 3

For this example we take the convection-diffusion coefficients $c = 0.5$, $d = 0.001$ and $\mu = 1/4$ for the problem (1)–(3) and $f(x, t)$ is chosen so that the exact solution is given by [8]

$$u(x, t) = 2(t^2 + t + 1) \sin^2 \pi x, \quad 0 \leq x \leq 1, \quad t > 0. \quad (23)$$

Table 10: Error norms using $M = 100$

δt	N	CTBS-DQ		CTBS-CT	
		E_∞	E_2	E_∞	E_2
10^{-4}	10	$1.600e-03$	$2.320e-04$	$4.825e-04$	$1.038e-04$
	50	$5.183e-04$	$4.843e-05$	$5.103e-04$	$4.814e-05$
	100	$5.186e-04$	$3.432e-05$	$5.106e-04$	$3.404e-05$
	200	$5.187e-04$	$2.432e-05$	$5.108e-04$	$2.406e-05$
	500	$5.188e-04$	$1.540e-05$	$5.108e-04$	$1.521e-05$
10^{-5}	10	$1.683e-04$	$2.300e-05$	$7.482e-06$	$1.638e-06$
	50	$3.700e-05$	$1.211e-06$	$8.995e-06$	$8.503e-07$
	100	$2.136e-05$	$6.500e-07$	$9.029e-06$	$6.026e-07$
	200	$1.296e-05$	$4.378e-07$	$9.037e-06$	$4.263e-07$
	500	$9.191e-06$	$2.738e-07$	$9.039e-06$	$2.696e-07$

Table 11: Comparison of error norms with CBSC method [8] using $N = 100$, $\delta t = 10^{-5}$, $P = 5$

M	CTBS-DQ		CTBS-CT		CBSC	
	E_∞	E_2	E_∞	E_2	E_∞	E_2
10	$2.502e-06$	$2.893e-08$	$1.685e-07$	$1.130e-08$	$6.616e-09$	$3.941e-10$
50	$1.159e-05$	$2.203e-07$	$2.687e-06$	$1.796e-07$	$3.308e-08$	$1.970e-09$
100	$2.136e-05$	$6.500e-07$	$9.029e-06$	$6.026e-07$	$6.618e-08$	$3.942e-09$
500	$1.527e-04$	$1.013e-05$	$1.517e-04$	$1.010e-05$	$3.314e-07$	$1.973e-08$

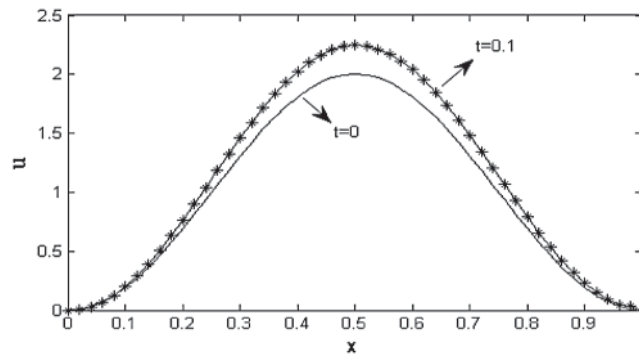


Figure 8: Numerical solutions obtained by CTBS-CT, using $N = 50$, $\delta t = 10^{-4}$, $c = 0.5$, $d = 0.001$, $\mu = 1/4$ at $t = 0, 0.1$

The conditions g , g_1 , g_2 are found from Eq. (23). The error norms for different values of N , δt and M are computed which are provided in Tabs. 10 and 11. It can be noted from Tabs. 10 and 11 that CTBS-CT provided better results than CTBS-DQ. The results in Tab. 11 correspond to convection dominated problem for $P=5$. Moreover, the results of the present methods are also compared with the method CBSC [8]. Fig. 8 shows solutions obtained using CTBS-CT at $t=0$, 0.1. Fig. 9 shows the solution obtained by CTBS-DQ at various time levels.

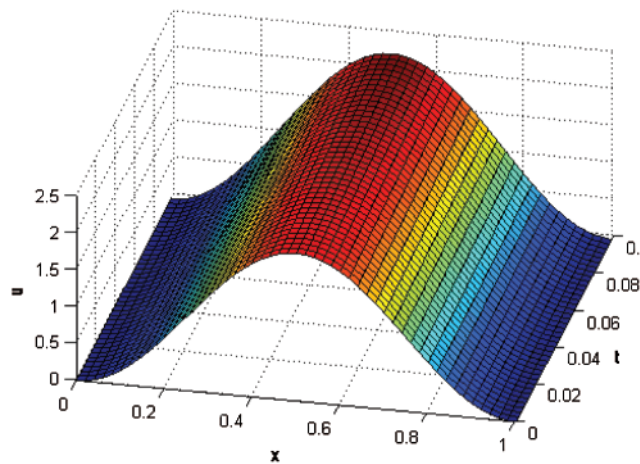


Figure 9: Numerical solutions obtained by CTBS-DQ at different times in the interval $[0, 0.1]$ using $N=50$, $\delta t=10^{-4}$, $c=0.5$, $d=0.001$, $\mu=1/4$

5 Conclusion

Two cubic trigonometric B-spline functions based methods are constructed for comparative study of approximate solution of a parabolic type partial integro-differential equation with a weakly singular kernel. The methods are validated using three test problems and the results are compared via error norms. The methods are computationally efficient and improved accuracy is obtained for relatively larger number of spatial nodes. Stability of the methods is shown via spectral radius. It is also observed that condition numbers of the system matrix obtained from cubic trigonometric B-spline collocation method does not increase with increase in the number of spatial nodes and hence the computation remained stable. Both methods do not require much smaller time step to attain high accuracy. In most cases, it is found that cubic trigonometric B-spline collocation method provides better accuracy as compared to cubic trigonometric B-spline differential quadrature. Also, in two examples the cubic trigonometric B-spline collocation technique provided better accuracy than cubic B-spline collocation method. Both techniques have the ability to solve convection dominated and diffusion dominated problems. Due to excellent agreement of these methods with the exact solution, the proposed techniques are efficient in employing to get approximate solution of PIDEs.

Acknowledgement: The authors are thankful to the reviewers for their helpful comments which improved the work of this paper.

Funding Statement: The author(s) received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Gurtin, M. E., Pipkin, A. C. (1968). A general theory of heat conduction with finite wave speeds. *Archive for Rational Mechanics and Analysis*, 31(2), 113–126. DOI 10.1007/BF00281373.
2. Pao, C. V., Payne, L., Amann, H. (2007). Bifurcation analysis of a nonlinear diffusion system in reactor dynamics. *Applicable Analysis*, 9(2), 107–119. DOI 10.1080/00036817908839258.
3. Zadeh, K. S. (2011). An integro-partial differential equation for modeling biofluids flow in fractured biomaterials. *Journal of Theoretical Biology*, 273(1), 72–79. DOI 10.1016/j.jtbi.2010.12.039.
4. Hepperger, P. (2012). Hedging electricity swaptions using partial integro-differential equations. *Stochastic Processes and their Applications*, 122(2), 600–622. DOI 10.1016/j.spa.2011.09.005.
5. Lee, Y. (2014). Financial options pricing with regime-switching jump-diffusions. *Computers & Mathematics with Applications*, 68(3), 392–404. DOI 10.1016/j.camwa.2014.06.015.
6. Larsson, S., Racheva, M., Saedpanah, F. (2015). Discontinuous Galerkin method for an integro-differential equation modeling dynamic fractional order viscoelasticity. *Computer Methods in Applied Mechanics and Engineering*, 283, 196–209. DOI 10.1016/j.cma.2014.09.018.
7. Mirzaee, F., Alipour, S. (2020). A hybrid approach of nonlinear partial mixed integro-differential equations of fractional order. *Iranian Journal of Science and Technology, Transactions A: Science*, 44(3), 725–737. DOI 10.1007/s40995-020-00859-7.
8. Siddiqi, S. S., Arshed, S. (2013). Numerical solution of convection-diffusion integro-differential equations with a weakly singular kernel. *Journal of Basic and Applied Science Research*, 3(11), 106–120.
9. Ali, A., Ahmad, S., Shah, S. I. A., Haq, F. (2015). A quartic B-spline collocation technique for the solution of partial integro-differential equations with a weakly singular kernel. *Science International*, 27(4), 2953–2958.
10. Fahim, A., Araghi, M. A. F. (2018). Numerical solution of convection-diffusion equations with memory term based on sinc method. *Computational Methods for Differential Equations*, 6(3), 380–395.
11. Al-Humedi, H. O., Jameel, Z. A. (2020). Combining cubic B-spline Galerkin method with quadratic weight function for solving partial integro-differential equations. *Journal of Al-Qadisiyah for Computer Science and Mathematics*, 12, 9–20.
12. Zhao, Y., Zhao, F. (2016). The analytical solution of parabolic volterra integro-differential equations in the infinite domain. *Entropy*, 18(10), 344. DOI 10.3390/e18100344.
13. Lopez-Marcos, J. C. (1990). A difference scheme for a nonlinear partial integro-differential equation. *SIAM Journal on Numerical Analysis*, 27(1), 20–31. DOI 10.1137/0727002.
14. Qiao, L., Xu, D., Wang, Z. (2019). An ADI difference scheme based on fractional trapezoidal rule for fractional integro-differential equation with a weakly singular kernel. *Applied Mathematics and Computation*, 354, 103–114. DOI 10.1016/j.amc.2019.02.022.
15. Chen, C., Thomee, V., Wahlbin, L. B. (1992). Finite element approximation of a parabolic integro-differential equation with a weakly singular kernel. *Mathematics of Computation*, 58(198), 587–602. DOI 10.1090/S0025-5718-1992-1122059-2.
16. Avazzadeh, Z., Rizi, Z. B., Ghaini, F. M. M., Loghmani, G. B. (2012). A numerical solution of nonlinear parabolic-type Volterra partial integro-differential equations using radial basis functions. *Engineering Analysis with Boundary Elements*, 36(5), 881–893. DOI 10.1016/j.enganabound.2011.09.013.
17. Fakhar-Izadi, F., Dehghan, M. (2011). The spectral methods for parabolic Volterra integro-differential equations. *Journal of Computational and Applied Mathematics*, 235(14), 4032–4046. DOI 10.1016/j.cam.2011.02.030.
18. Long, W., Xu, D., Zeng, X. (2012). Quasi wavelet based numerical method for a class of partial integro-differential equation. *Applied Mathematics and Computation*, 218(24), 11842–11850. DOI 10.1016/j.amc.2012.04.090.

19. Ali, A., Khan, K., Haq, F., Shah, S. I. A. (2019). A computational modeling based on trigonometric cubic B-spline functions for the approximate solution of a second order partial integro-differential equation. *New Knowledge in Information Systems and Technologies*, vol. 930, AISC, Spain: Springer Nature Switzerland.
20. Mirzaee, F., Alipour, S. (2019). An efficient cubic B-spline and bicubic B-spline collocation method for numerical solutions of multidimensional nonlinear stochastic quadratic integral equations. *Mathematical Methods in the Applied Sciences*, 43(1), 384–397. DOI 10.1002/mma.5890.
21. Qiao, L., Wang, Z., Xu, D. (2020). An alternating direction implicit orthogonal spline collocation method for the two dimensional multi-term time fractional integro-differential equation. *Applied Numerical Mathematics*, 151, 199–212. DOI 10.1016/j.apnum.2020.01.003.
22. Abbas, M., Majid, A. A., Ismail, A. I. M., Rashid, A. (2014). The application of cubic trigonometric B-spline to the numerical solution of the hyperbolic problems. *Applied Mathematics and Computation*, 239, 74–88. DOI 10.1016/j.amc.2014.04.031.
23. Irk, D., Keskin, P. (2016). Cubic trigonometric B-spline Galerkin methods for the regularized long wave equation. *Journal of Physics: Conference Series*, 766, 12032. DOI 10.1088/1742-6596/766/1/012032.
24. Dag, I., Hepson, O. E., Kacmaz, O. (2017). The trigonometric cubic B-spline algorithm for Burgers' equation. *International Journal of Nonlinear Science*, 24(2), 120–128.
25. Hashmi, M. S., Awais, M., Waheed, A., Ali, Q. (2017). Numerical treatment of Hunter Saxton equation using cubic trigonometric B-spline collocation method. *AIP Advances*, 7(9), 95124. DOI 10.1063/1.4996740.
26. Yaseen, M., Abbas, M., Nazir, T., Baleanu, D. (2017). A finite difference scheme based on cubic trigonometric B-splines for a time fractional diffusion-wave equation. *Advances in Difference Equations*, 2017(1), 291. DOI 10.1186/s13662-017-1330-z.
27. Korkmaz, A., Akmaz, H. K. (2018). Numerical solution of non-conservative linear transport problems. *Journal of Applied and Engineering Mathematics*, 8(1a), 167–177.
28. Tamsir, M., Dhiman, N., Srivastava, V. K. (2018). Cubic trigonometric B-spline differential quadrature method for numerical treatment of Fisher's reaction-diffusion equations. *Alexandria Engineering Journal*, 57(3), 2019–2026. DOI 10.1016/j.aej.2017.05.007.
29. Schoenberg, I. J. (1964). On trigonometric spline interpolation. *Journal of Mathematics and Mechanics*, 13, 795–825.
30. Koch, P. E., Lyche, T., Neamtu, M., Schumaker, L. L. (1995). Control curves and knot insertion for trigonometric splines. *Advances in Computational Mathematics*, 3(4), 405–424. DOI 10.1007/BF03028369.
31. Walz, G. (1997). Identities for trigonometric B-splines with an application to curve design. *BIT Numerical Mathematics*, 37(1), 189–201. DOI 10.1007/BF02510180.
32. Lyche, T., Winther, R. (1979). A stable recurrence relation for trigonometric B-splines. *Journal of Approximation Theory*, 25(3), 266–279. DOI 10.1016/0021-9045(79)90017-0.
33. Hamid, N. N. A., Majid, A. A., Ismail, A. I. M. (2010). Cubic trigonometric B-spline applied to linear two-point boundary value problems of order two. *World Academy of Science, Engineering and Technology International Journal of Mathematical and Computational Sciences*, 4(10), 1377–1382.
34. Siraj-ul-Islam Ali, A., Zafar, A., Hussain, A. (2020). A differential quadrature based approach for Volterra partial integro-differential equation with a weakly singular kernel. *Computer Modeling in Engineering & Sciences*, 124(3), 915–935. DOI 10.32604/cmescs.2020.011218.
35. Abbas, M., Majid, A. A., Ismail, A. I. M., Rashid, A. (2014). Numerical method using cubic trigonometric B-spline technique for nonclassical diffusion problems. *Abstract and Applied Analysis*, 2014(1–6), 1–11. DOI 10.1155/2014/849682.
36. Nazir, T., Abbas, M., Yaseen, M. (2017). Numerical solution of second-order hyperbolic telegraph equation via new cubic trigonometric B-splines approach. *Cogent Mathematics*, 4(1), 1382061. DOI 10.1080/23311835.2017.1382061.
37. Hepson, O. E., Dag, I. (2017). The numerical approach to the Fisher's equation via trigonometric cubic B-spline collocation method. *Communications in Numerical Analysis*, 2017(2), 91–100. DOI 10.5899/2017/cna-00293.

38. Kumar Singh, B., Kumar, P. (2018). An algorithm based on DQM with modified trigonometric cubic B-splines for solving coupled viscous Burgers' equations. *Communications in Numerical Analysis*, 2018(1), 21–41. DOI 10.5899/2018/cna-00333.
39. Frazer, R. A., Jones, W. P., Skan, S. W. (1937). *ARC R and M 1799*. New York: Springer-Verlag New York Inc.
40. Zhang, X., Liu, X. H., Song, K. Z., Lu, M. W. (2001). Least-squares collocation meshless method. *International Journal for Numerical Methods in Engineering*, 51(9), 1089–1100. DOI 10.1002/nme.200.
41. Hsiao, G. C., Kopp, P., Wendland, W. L. (1980). A Galerkin collocation method for some integral equations of the first kind. *Computing*, 25(2), 89–130. DOI 10.1007/BF02259638.
42. Quan, J. R., Chang, C. T. (1989). New insights in solving distributed system equations by the quadrature method I. Analysis. *Computers and Chemical Engineering*, 13(7), 779–788. DOI 10.1016/0098-1354(89)85051-3.
43. Bert, C. W., Malik, M. (1996). Differential quadrature method in computational mechanics: A review. *Applied Mechanics Reviews*, 49(1), 1–28. DOI 10.1115/1.3101882.
44. Bellman, R. E., Casti, J. (1971). Differential quadrature and long-term integration. *Journal of Mathematical Analysis and Applications*, 34(2), 235–238. DOI 10.1016/0022-247X(71)90110-7.
45. Bellman, R. E., Kashef, B. G., Casti, J. (1972). Differential quadrature: A technique for the rapid solution of nonlinear partial differential equations. *Journal of Computational Physics*, 10(1), 40–52. DOI 10.1016/0021-9991(72)90089-7.
46. Shu, C. (2000). *Differential quadrature and its application in engineering*. London: Springer-Verlag.
47. Ahmad, I., Ahsan, M., Hussain, I., Kumam, P., Kumam, W. (2019). Numerical simulation of PDEs by local meshless differential quadrature collocation method. *Symmetry*, 11(3), 394. DOI 10.3390/sym11030394.
48. Khan, M. N., Siraj-ul-Islam, Hussain, I., Ahmad, I., Ahmad, H. (2020). A local meshless method for the numerical solution of space-dependent inverse heat problems. *Mathematical Methods in the Applied Sciences*, 2020, 1–14.
49. Bashan, A. (2019). An efficient approximation to numerical solutions for the Kawahara equation via modified cubic B-spline differential quadrature method. *Mediterranean Journal of Mathematics*, 16(1), 708717. DOI 10.1007/s00009-018-1291-9.
50. Korkmaz, A., Dag, I. (2011). Shock wave simulations using Sinc differential quadrature method. *Engineering Computations*, 28(6), 654–674. DOI 10.1108/02644401111154619.
51. Arora, G., Joshi, V. (2018). A computational approach using modified trigonometric cubic B-spline for numerical solution of Burgers' equation in one and two dimensions. *Alexandria Engineering Journal*, 57(2), 1087–1098. DOI 10.1016/j.aej.2017.02.017.
52. Ghafoor, A., Haq, S., Hussain, M., Kumam, P., Jan, M. A. (2019). Approximate solutions of time fractional diffusion wave models. *Mathematics*, 7(10), 923. DOI 10.3390/math7100923.
53. Siraj-ul-Islam, Haq, S., Ali, A. (2009). A meshfree method for the numerical solution of the RLW equation. *Journal of Computational and Applied Mathematics*, 223(2), 997–1012. DOI 10.1016/j.cam.2008.03.039.