

International Journal of Mathematical Modelling and Numerical Optimisation

ISSN online: 2040-3615 - ISSN print: 2040-3607

<https://www.inderscience.com/ijmmno>

Non-polynomial cubic triplet parameter spline scheme for second-order boundary value problem systems

Pankaj Kumar Srivastava

DOI: [10.1504/IJMMNO.2025.10069808](https://doi.org/10.1504/IJMMNO.2025.10069808)

Article History:

Received:	20 December 2024
Last revised:	02 February 2025
Accepted:	07 February 2025
Published online:	08 April 2025

Non-polynomial cubic triplet parameter spline scheme for second-order boundary value problem systems

Pankaj Kumar Srivastava

Department of Mathematics,
Jaypee Institute of Information Technology,
Noida, India
Email: pankaj.srivastava@jiiit.ac.in

Abstract: Understanding second-order boundary value problems (BVPs) aids in modelling and predicting various real-world phenomena, crucial for scientific and engineering advancements. This study proposes the development of an effective numerical method based on cubic splines and non-polynomial triplet parameters for approximations to the solution of systems of second-order BVPs. This method extends the non-polynomial cubic spline's trigonometric part by three additional parameters, which makes it superior to other existing numerical techniques. The cherry on top is the suggested scheme's adaptability to different step sizes. With the inclusion of CPU time, this analysis gets more fascinating. Comparatively speaking to alternative spline, collocation, and finite difference approaches, the current algorithm provides better approximations. To build a solid foundation for the suggested method, the convergence analysis of the proposed algorithm is discussed. Numerical examples are used to demonstrate the suggested method's utility in practice.

Keywords: non-polynomial triplet parameter spline; differential equations; convergence analysis; maximum absolute error; MAE; central processing unit time.

Reference to this paper should be made as follows: Srivastava, P.K. (2025) 'Non-polynomial cubic triplet parameter spline scheme for second-order boundary value problem systems', *Int. J. Mathematical Modelling and Numerical Optimisation*, Vol. 15, No. 1, pp.27–51.

Biographical notes: Pankaj Kumar Srivastava, a theoretical mathematician with 20+ years of experience, specialises in fuzzy optimisation and computational methods. He has taught at institutions under Dr. A.P.J. Abdul Kalam Technical University and Jaypee Institute of Information Technology, where he is an Associate Professor. With more than 55 publications in top journals and conferences, he has guided four PhD scholars and reviewed esteemed journals. A member of multiple mathematical societies, he also serves on editorial boards. Dedicated to learner-centric education, he fosters critical thinking and has organised conferences, faculty development programs, and workshops to promote academic and research excellence.

1 Introduction

Due to its numerous uses in scientific study, second-order boundary value problem (SO-BVP) systems' approximative solutions are actually of utmost relevance. In addition

to the challenges in finding precise solutions, numerical algorithms are becoming more relevant in mathematical applications since they can readily be used in conjunction with modern, fast digital computers. Such systems are numerous, and it is difficult to use straightforward methods to find their solutions. The literature contains a variety of techniques for solving SO-BVP systems. This study focuses on numerical solution of second-order boundary value problem (BVP) systems, which are widely used in mathematical modelling of real-world issues, applied sciences, engineering, and theoretical physics.

There are several approaches in the literature for solving SO-BVP systems. Caglar and Caglar (2009) employed the B-spline approach to crack the linear system of SO-BVPs, while Cheng and Zhong (2005) solved the second-order ordinary differential system discussed by discussion of the presence of positive solutions. Thompson and Tisdell (2000) introduced difference equation systems, and Dehghan and Saadatmandi (2007) invented the sinc-collocation approach. There are not many methods in the literature for finding approximations of solutions to nonlinear systems. El-Gamel (2012) used the sinc-collocation method to solve nonlinear and singly perturbed SO-BVP systems. Lu (2007) invented the variational iteration approach for solving a nonlinear system, while Geng and Cui (2007) invented the reproducing kernel space method. While Srivastava and Kumar (2012) employed quintic non-polynomial spline to solve third-order BVPs connected to draining and coating flows, Bataineh et al. (2009) offered a modified homotopy technique for systems of SO-BVPs. Srivastava et al. (2011, 2012) used non-polynomial quintic splines to solve second- and fourth-order BVPs.

In the modern period, spline-based solutions have become a reliable resource for addressing a wide range of issues. It is piecewise polynomial property gives it a competitive edge over existing numerical approaches. The splines are categorised into certain type based on order: linear, quadratic, cubic, and of higher orders (Srivastava, 2015). It is classified as non-polynomial or polynomial depending on whether trigonometric functions are present or absent. It is possible to change the non-polynomial spline into a polynomial spline (Srivastava et al., 2012; Kumar and Srivastava, 2008, 2009, 2011; Srivastava, 2014). Non-polynomial splines are infinitely differentiable due to the inclusion of trigonometric functions, which contributes to more precise problem-solving. The existence of a free parameter in the trigonometric portion is the primary characteristic of non-polynomial splines. When solving problems involving empirical mode decomposition, this parameter acts as a frequency (Srivastava and Kumar, 2012; Singh et al., 2013). Additionally, differential equation model approximation has long been a topic of study. That contains some reliable work on the solution of SO-BVPs. The most efficient methods for handling SO-BVPs include high-order generalisations of traditional finite difference schemes on the first and second derivatives (Amodio and Sgura, 2005), spectral-homotopy analysis (Motsa et al., 2010), continuous genetic algorithm (Abu-Arquib et al., 2014), a set of guidelines for recreating kernel Hilbert space (Sahihi et al., 2020), and improved Legendre orthonormal basis (Sun et al., 2021). This study is based on the resolution of generalised second order boundary value problem systems by non-polynomial cubic triplet parameter spline scheme. The system under consideration is of the form as follows:

$$\frac{d^2 y}{dx^2} = m(x, y, z) \frac{dz}{dx} + l(x, y, z) \frac{dy}{dx} + y(x)n(x, y, z) + z(x)p(x, y, z) + q(x, y, z) \quad (1.1)$$

$$\frac{d^2z}{dx^2} = g(x, y, z) \frac{dz}{dx} + f(x, y, z) \frac{dy}{dx} + y(x)h(x, y, z) + z(x)\mathcal{O}(x, y, z) + \varphi(x, y, z) \quad (1.2)$$

with

$$y(a) = \vartheta_0, y(b) = \vartheta_n, y(c) = \eta_0, y(d) = \eta_n \quad (1.3)$$

where $l(x, y, z)$, $m(x, y, z)$, $n(x, y, z)$, $p(x, y, z)$, $q(x, y, z)$, $f(x, y, z)$, $g(x, y, z)$, $h(x, y, z)$, $\mathcal{O}(x, y, z)$ and $\varphi(x, y, z)$ are continuous functions in the interval $y \in [a, b]$, and $z \in [c, d]$ and ϑ_0 , ϑ_n , η_0 , η_n are constants.

At the beginning of the 21st century, the numerical treatment of SO-BVPs occurring in contact, unilateral, and obstruction scenarios was effectively completed. These problems have been connected to the problem of the equilibrium of elastic bodies in contact with a rigid base and can be investigated within the context of variational inequality theory. In the formulation of the theory of variational inequality, the approximation of second order differential equation models becomes an integral part of the solution. Uniform cubic splines were successfully used to manage it, and they were proven to be superior to other finite difference and collocation approaches (Al-Said, 2001). With the help of quadratic, cubic, and non-polynomial cubic spline one parameter functions, Ramadan et al. (2007) later covered it and established a solid foundation for the use of spline technology to resolve SO-BVPs. Further this method has been expanded to handle such BVPs using a non-polynomial quintic one parameter approach (Srivastava, 2019). Additionally, Liu et al. (2011) created a difference scheme grounded on quartic splines to approximate Neumann-type boundary conditions based both linear and nonlinear SO-BVPs.

Similar issues were resolved using a collocation technique based on quintic B-spline by Lang and Xu (2012). Furthermore, Cabada, Enguic, and Somoza examined the chances of discovering positive-solutions for SO-BVPs with interrupted and Dirichlet boundary conditions. In response to these circumstances, the Green's functions change sign. These outcomes held true for mixed circumstances as well (Cabada Fernández et al., 2017). Arqub et al. replicated kernel theory under the assumption of generalised differentiability to explore the analytical solutions as well as the second-order two-point fuzzy BVPs' approximation solutions. Utilising the learned kernel functions, the process also included the creation of orthogonal bases (Arqub et al., 2017). The block Falkner-style technique which is three-time differentiable was developed to solve SO-BVPs subject to each potential boundary condition. The method was created using the interpolation and collocation approaches (Ramos and Rufai, 2019). Zheng et al. (2020) created a multi-scale technique to resolve the SO-BVPs in a recent paper.

In addition to these, recent developments in numerical techniques for solving SO-BVPs have contributed significantly to numerous engineering and scientific fields. Karthick and Subburayan (2022) developed finite difference methods with linear interpolation for solving coupled hyperbolic delay differential equations, improving numerical stability. Achhab et al. (2023) introduced an adaptive finite element method for Wick stochastic partial differential equations, which enhances solution adaptability in stochastic environments. Rawat et al. (2023) focused on the convergence analysis and an efficient numerical technique for the Benjamin Bona Mahony equation, ensuring improved accuracy. Similarly, Sriwastav and Barnwal (2023) proposed a numerical solution for Lane-Emden pantograph delay differential equations, emphasising stability

and convergence. Das et al. (2022) developed a second-order weighted monotone numerical scheme for time-delayed parabolic initial-boundary-value problems, handling small perturbations effectively. Güngör and Altun (2024) explored new fixed-point theorems on quasi-metric spaces to establish existence results for SO-BVPs. Erdogan (2024) introduced a second-order numerical method for singularly perturbed Volterra integro-differential equations with delay, offering high accuracy. Sabir et al. (2024) integrated a neuro-evolution heuristic with sequential quadratic programming for second-order prediction differential models, combining machine learning with numerical analysis. Zhang (2023) analysed the oscillation behaviour of certain second-order linear differential equations, providing deeper theoretical insights. Adeyanju (2023) proposed stability and boundedness criteria for solutions of specific second-order differential equations, contributing to the fundamental understanding of system dynamics. Ala'yed et al. (2023) applied cubic B-splines to solve the quadratic Riccati differential equation, demonstrating the method's efficiency. Parumasur et al. (2023) explored the use of B-spline approximation functions for solving Burgers' and KdV-Burgers' equations, highlighting their applicability in fluid dynamics. Masti and Sayevand (2024) introduced a collocation-Galerkin method using fractional B-spline functions for stochastic fractional integro-differential equations. Jena and Gebremedhin (2023) developed a decatic B-spline collocation scheme for Burgers' equations, achieving higher-order accuracy. Babu and Asharaf (2023) utilised modified cubic B-spline-based differential quadrature methods to solve the nonlinear Sine-Gordon equation efficiently. Negero (2023) proposed a fitted cubic spline in tension difference scheme for singularly perturbed delay parabolic equations, ensuring solution stability. Lubo and Duressa (2023) employed linear B-spline finite element methods for solving delay reaction-diffusion equations, showcasing their effectiveness. Arumugam et al. (2025) presented a quintic spline-based computational method for singularly perturbed periodic BVPs, improving periodic solutions' accuracy. Daba et al. (2025) investigated singularly perturbed time-lag parabolic differential-difference equations numerically, addressing time-delay challenges. Palav and Pradhan (2025) developed a fourth-order hyperbolic polynomial B-spline collocation method for the advection-diffusion equation, improving numerical efficiency. Mangal and Gupta (2025) introduced an exponential B-spline collocation method with Richardson extrapolation for solving the generalised Black-Scholes equation, advancing financial modelling techniques. Finally, an additional 2025 study by Hariharan et al. (2025) proposed a dynamic operational matrix approach for nonlinear second-order differential equations, enhancing both accuracy and computational stability. Their method integrates finite volume discretisation with dynamic operational matrix, making it particularly effective for problems involving sharp gradients and singular perturbations. These studies collectively advance numerical methods for SO-BVPs, addressing various theoretical and practical challenges across multiple disciplines.

In addition to all of previous studies, the current work represents an effective attempt to address a system of SO-BVPs using a non-polynomial triplet parameter spline-based approach. The coverage is broken down into sections that cover the definition of the non-polynomial triplet parameter cubic spline, the method's derivation, its description, and the setting up of border circumstances. The remainder of the section includes convergence investigation, numerical examples, discussion and a comparative study, establishment of theorem on proposed study with real word applications including the conclusion at the end.

2 Cubic non-polynomial triplet parameter spline

The process of spline interpolation creates a piecewise function that roughly model's boundary value issues. A vector space in the range $[U, V]$ and a continued knot vector in the range are represented by the degree m spline. As an illustration, the C^2 cubic spline space contains the subspace of the natural cubic spline space (Srivastava, 2014; Srivastava et al., 2012).

A region-specific piecewise polynomial function W such that W may be divided into smaller parts is often considered to be a polynomial spline of degree m , each of which has a polynomial function of degree m . Additionally, it has derivatives of order up to $(m - 1)$ and is a continuous function in W .

In this study, the proposed term 'non-polynomial triplet parameter spline' refers to spline functions with triplet parameters (ρ_1, ρ_2, ρ_3) that are described in every subinterval of the domain in order to effectively handle boundary value concerns. Assured smoothness criteria at the knots are preferred to arbitrary constants. If the parameter is $(\rho_1, \rho_2, \rho_3) \rightarrow (0, 0, 0)$, such 'splines' arranged in the class $C^\infty[U, V]$ are changed to polynomial splines. The method used to introduce the free parameters determines the spline's precise appearance.

The following is the structure of the projected spline functions in this article

$$T = \text{span}\{1, \sin(\rho_1 w), \cos(\rho_2 w), \sin(\rho_3 w)\}$$

The non-polynomial spline's C^∞ -differentiability compensates for the polynomial spline's loss of smoothness. Once the correlation between the base of polynomial and non-polynomial splines has been examined using the following theme:

$$\lim_{(\rho_1, \rho_2, \rho_3) \rightarrow (0, 0, 0)} T = \lim_{(\rho_1, \rho_2, \rho_3) \rightarrow (0, 0, 0)} \text{span}\left\{1, \frac{1}{\rho_1} \sin(\rho_1 w), \frac{2}{\rho_2^2} [1 - \cos(\rho_2 w)], \frac{1}{(\rho_3)^3} [\rho_3 t - \sin(\rho_3 w)]\right\} \quad (2.1)$$

The equation (2.1) contents

$$\lim_{(\rho_1, \rho_2, \rho_3) \rightarrow (0, 0, 0)} T = \lim_{(\rho_1, \rho_2, \rho_3) \rightarrow (0, 0, 0)} \text{span}\{1, w, w^2, w^3\} \quad (2.2)$$

where (ρ_1, ρ_2, ρ_3) are the spline function's trigonometric frequencies, which might be either pure real or imaginary. These have been used to increase the scheme's accuracy. Compared to finite difference approaches, a continuous approximation to the function $p(z)$ has been provided and its advanced derivatives.

3 Development of the scheme

In order to get close to the solution of the under consideration second order differential equation, the mathematical scheme has been shaken up by taking a mesh Δ and nodes w_j on $[U, V]$ such that

$$\Delta : U = w_0 < w_1 < w_2 < w_3 \dots < w_n = V$$

having $w_j = U + jh_j, j = 0, 1, 2, \dots, n$, where $h_j = w_{j+1} - w_j, j = 0, 1, \dots, n - 1$.

The non-polynomial function $S_j(w)$ belongs to $C^\infty[U, V]$ and interpolating $y(w)$ at the node arguments $w_j, j = 0, 1, 2, \dots, n$, which hinges on three independent parameters (ρ_1, ρ_2, ρ_3) , and converted to usual cubic spline $S_j(w)$ in $[u, v]$ as $(\rho_1, \rho_2, \rho_3) \rightarrow (0, 0, 0)$.

In every section $[w_j, w_{j+1}], j = 0, 1, 2, \dots, n - 1$, the non-polynomial $S_j(w)$ is expressed by

$$S_j(w) = a_j + b_j \text{Sin} \rho_1 (w - w_j) + c_j \text{Cos} \rho_2 (w - w_j) + d_j \text{Sin} \rho_3 (w - w_j) \quad (3.1)$$

where a_j, b_j, c_j , and d_j are constants with ρ_1, ρ_2, ρ_3 independent parameters.

Let y_j be a close approximation to $y(w_j)$, which is obtained by the spline function's section $S_j(w)$, which passes through (w_j, y_j) and (w_{j+1}, y_{j+1}) . Additionally, in order to get the appropriate requirements for the coefficients established in equation (3.1), the $S_j(w)$ needs to interpolate w_j and w_{j+1} , as well as have a continuous first derivative at the common knots (w_j, y_j) .

When the coefficients of equation (3.1) are expressed in terms of y_j, y_{j+1}, M_j and M_{j+1} , the following first denotes:

$$S_j(w_j) = y_j, S_j(w_{j+1}) = y_{j+1},$$

and

$$S_j''(w_j) = M_j, S_j''(w_{j+1}) = M_{j+1}, j = 0, 1, 2, \dots, n - 1.$$

From equation (3.1)

$$S_j(w_j) = a_j + c_j = y_j$$

$$S_j(w_{j+1}) = a_j + b_j \text{Sin} \rho_1 h_j + c_j \text{Cos} \rho_2 h_j + d_j \text{Sin} \rho_3 h_j \quad (3.2)$$

$$S_j'(w) = b_j \rho_1 \text{Cos} \rho_1 (w - w_j) - c_j \rho_2 \text{Sin} \rho_2 (w - w_j) + d_j \rho_3 \text{Cos} \rho_3 (w - w_j)$$

$$\begin{aligned} S_j'(w_{j+1}) &= b_j \rho_1 \text{Cos} \rho_1 h_j - c_j \rho_2 \text{Sin} \rho_2 h_j + d_j \rho_3 \text{Cos} \rho_3 h_j = S_{j+1}'(w_{j+1}) \\ &= b_{j+1} \rho_1 + d_{j+1} \rho_3 \end{aligned} \quad (3.3)$$

$$S_j''(w) = -b_j \rho_1^2 \text{Sin} \rho_1 (w - w_j) - c_j \rho_2^2 \text{Cos} \rho_2 (w - w_j) - d_j \rho_3^2 \text{Sin} \rho_3 (w - w_j)$$

$$S_j''(w_{j+1}) = S_{j+1}''(w_{j+1}) \quad (3.4)$$

It implies,

$$-b_j \rho_1^2 \text{Sin} \rho_1 h_j - c_j \rho_2^2 \text{Cos} \rho_2 h_j - d_j \rho_3^2 \text{Sin} \rho_3 h_j = -c_{j+1} \rho_2^2 \quad (3.5)$$

$$M_j = S_j''(w_j) = -c_j \rho_2^2 \quad (3.6)$$

It gives

$$(i) c_j = -\frac{M_j}{\rho_2^2}.$$

$$(ii) a_j = y_j + \frac{M_j}{\rho_2^2}.$$

$$M_{j+1} = S_j''(w_{j+1}) = -b_j \rho_1^2 \text{Sin} \rho_1 h_j - c_j \rho_2^2 \text{Cos} \rho_2 h_j - d_j \rho_3^2 \text{Sin} \rho_3 h_j \quad (3.7)$$

It gives,

$$-b_j \rho_1^2 \text{Sin} \rho_1 h_j - d_j \rho_3^2 \text{Sin} \rho_3 h_j = M_{j+1} + c_j \rho_2^2 \text{Cos} \rho_2 h_j \quad (3.8)$$

From equation (3.1)

$$b_j \text{Sin} \rho_1 h_j + d_j \text{Sin} \rho_3 h_j = y_{j+1} - a_j - c_j \text{Cos} \rho_2 h_j \quad (3.9)$$

From equations (3.8) and (3.9) using cross multiplication

$$b_j = \frac{1}{X} \begin{vmatrix} y_{j+1} - a_j - c_j \text{Cos} \rho_2 h_j & \text{Sin} \rho_3 h_j \\ M_{j+1} + c_j \rho_2^2 \text{Cos} \rho_2 h_j & -\rho_3^2 \text{Sin} \rho_3 h_j \end{vmatrix}$$

and

$$d_j = \frac{1}{X} \begin{vmatrix} \text{Sin} \rho_1 h_j & y_{j+1} - a_j - c_j \text{Cos} \rho_2 h_j \\ -\rho_1^2 \text{Sin} \rho_1 h_j & M_{j+1} + c_j \rho_2^2 \text{Cos} \rho_2 h_j \end{vmatrix}$$

where

$$X = (\rho_1^2 - \rho_3^2) \text{Sin} \rho_1 h_j \text{Sin} \rho_3 h_j.$$

On simplification

$$(iii) b_j = \frac{\left\{ -y_{j+1} \rho_3^2 + y_j \rho_3^2 + M_j \left(\frac{\rho_3}{\rho_2} \right)^2 - M_{j+1} - M_j \left[\left(\frac{\rho_3}{\rho_2} \right)^2 - 1 \right] \text{Cos} \rho_2 h_j \right\}}{(\rho_1^2 - \rho_3^2) \text{Sin} \rho_1 h_j}$$

and

$$(iv) d_j = \frac{\left\{ M_{j+1} + \text{Cos} \rho_2 h_j M_j \left(\frac{\rho_1^2}{\rho_2^2} - 1 \right) + \rho_1^2 \left(y_{j+1} - y_j - \frac{M_j}{\rho_2^2} \right) \right\}}{(\rho_1^2 - \rho_3^2) \text{Sin} \rho_3 h_j}$$

From equation (3.3) we have

$$b_j \rho_1 \text{Cos} \rho_1 h_j - c_j \rho_2 \text{Sin} \rho_2 h_j + d_j \rho_3 \text{Cos} \rho_3 h_j = b_{j+1} \rho_1 + d_{j+1} \rho_3 \quad (3.10)$$

Using values of b_j , b_{j+1} , c_j , d_j and b_{j+1} in equation (3.10), it becomes

$$\begin{aligned}
& \frac{\left\{ -y_{j+1}\rho_3^2 + y_j\rho_3^2 + M_j \left(\frac{\rho_3}{\rho_2} \right)^2 - M_{j+1} - M_j \left[\left(\frac{\rho_3}{\rho_2} \right)^2 - 1 \right] \text{Cosp}_2 h_j \right\}}{(\rho_1^2 - \rho_3^2) \text{Sin} \rho_1 h_j} \rho_1 \text{Cosp}_1 h_j \\
& + \frac{M_j}{\rho_2} \text{Sin} \rho_2 h_j + \frac{\left\{ M_{j+1} + \text{Cosp}_2 h_j M_j \left(\frac{\rho_1^2}{\rho_2^2} - 1 \right) + \rho_1^2 \left(y_{j+1} - y_j - \frac{M_j}{\rho_2^2} \right) \right\}}{(\rho_1^2 - \rho_3^2) \text{Sin} \rho_3 h_j} \rho_3 \text{Cosp}_3 h_j \\
& = \frac{\left\{ -y_{j+2}\rho_3^2 + y_{j+1}\rho_3^2 + M_{j+1} \left(\frac{\rho_3}{\rho_2} \right)^2 - M_{j+2} - M_{j+1} \left[\left(\frac{\rho_3}{\rho_2} \right)^2 - 1 \right] \text{Cosp}_2 h_{j+1} \right\}}{(\rho_1^2 - \rho_3^2) \text{Sin} \rho_1 h_{j+1}} \rho_1 \\
& + \frac{\left\{ M_{j+2} + \text{Cosp}_2 h_{j+1} M_{j+1} \left(\frac{\rho_1^2}{\rho_2^2} - 1 \right) + \rho_1^2 \left(y_{j+2} - y_{j+1} - \frac{M_{j+1}}{\rho_2^2} \right) \right\}}{(\rho_1^2 - \rho_3^2) \text{Sin} \rho_3 h_{j+1}} \rho_3
\end{aligned}$$

this implies that

$$\begin{aligned}
& \left[\left(M_j \left(\frac{\rho_3}{\rho_2} \right)^2 - M_{j+1} \right) + \rho_3^2 (y_j - y_{j+1}) - M_j \left[\left(\frac{\rho_3}{\rho_2} \right)^2 - 1 \right] \text{Cosp}_2 h_j \right] \rho_1 \text{Cot} \rho_1 h_j \\
& + \frac{M_j}{\rho_2} (\rho_1^2 - \rho_3^2) \text{Sin} \rho_2 h_j + \left[M_{j+1} + M_j \left(\frac{\rho_1^2}{\rho_2^2} - 1 \right) \text{Cosp}_2 h_j + \rho_1^2 \left(y_{j+1} - y_j - \frac{M_j}{\rho_2^2} \right) \right] \\
& \rho_3 \text{Cot} \rho_3 h_j = \rho_1 \frac{\left\{ \begin{aligned} & -y_{j+2}\rho_3^2 + y_{j+1}\rho_3^2 + M_{j+1} \left(\frac{\rho_3}{\rho_2} \right)^2 - M_{j+2} \\ & - M_{j+1} \left[\left(\frac{\rho_3}{\rho_2} \right)^2 - 1 \right] \text{Cosp}_2 h_{j+1} \end{aligned} \right\}}{\text{Sin} \rho_1 h_{j+1}} \\
& + \frac{\left\{ M_{j+2} + \text{Cosp}_2 h_{j+1} M_{j+1} \left(\frac{\rho_1^2}{\rho_2^2} - 1 \right) + \rho_1^2 \left(y_{j+2} - y_{j+1} - \frac{M_{j+1}}{\rho_2^2} \right) \right\}}{\text{Sin} \rho_3 h_{j+1}} \rho_3
\end{aligned}$$

so

$$\begin{aligned}
 & M_j \left[\frac{\rho_3^2}{\rho_2^2} \rho_1 \text{Cot} \rho_1 h_j - \left(\frac{\rho_3^2}{\rho_2^2} - 1 \right) \rho_1 \text{Cot} \rho_1 h_j \text{Cos} \rho_2 h_j + \left(\frac{\rho_1^2 - \rho_3^2}{\rho_2} \right) \text{Sin} \rho_2 h_j \right. \\
 & \left. + \left(\frac{\rho_1^2}{\rho_2^2} - 1 \right) \rho_3 \text{Cos} \rho_2 h_j \text{Cot} \rho_3 h_j - \frac{\rho_1^2}{\rho_2^2} \rho_3 \text{Cot} \rho_3 h_j \right] \\
 & + M_{j+1} \left[-\rho_1 \text{Cot} \rho_1 h_j + \rho_3 \text{Cot} \rho_3 h_j \right] \\
 & \rho_1 \left\{ M_{j+1} \left(\frac{\rho_3}{\rho_2} \right)^2 - M_{j+2} - M_{j+1} \left[\left(\frac{\rho_3}{\rho_2} \right)^2 - 1 \right] \text{Cos} \rho_2 h_{j+1} \right\} \\
 & - \frac{\text{Sin} \rho_1 h_{j+1}}{\text{Sin} \rho_3 h_{j+1}} \\
 & \rho \left\{ M_{j+2} + \text{Cos} \rho_2 h_{j+1} M_{j+1} \left(\frac{\rho_1^2}{\rho_2^2} - 1 \right) - \rho_1^2 \left(\frac{M_{j+1}}{\rho_2^2} \right) \right\} \\
 & - \frac{\text{Sin} \rho_3 h_{j+1}}{\text{Sin} \rho_1 h_{j+1}} \\
 & = \frac{\rho_1 \{-y_{j+2} \rho_3^2 + y_{j+1} \rho_3^2\}}{\text{Sin} \rho_1 h_{j+1}} + \frac{\rho_3 \{\rho_1^2 (y_{j+2} - y_{j+1})\}}{\text{Sin} \rho_3 h_{j+1}} \\
 & - \rho_1 \rho_3^2 (y_j - y_{j+1}) \text{Cot} \rho_1 h_j - \rho_1^2 \rho_3 (y_{j+1} - y_j) \text{Cot} \rho_3 h_j
 \end{aligned}$$

it becomes

$$\begin{aligned}
 & M_j \left[\frac{\rho_3^2}{\rho_2^2} \rho_1 \text{Cot} \rho_1 h_j - \left(\frac{\rho_3^2}{\rho_2^2} - 1 \right) \rho_1 \text{Cot} \rho_1 h_j \text{Cos} \rho_2 h_j + \left(\frac{\rho_1^2 - \rho_3^2}{\rho_2} \right) \text{Sin} \rho_2 h_j \right. \\
 & \left. + \left(\frac{\rho_1^2}{\rho_2^2} - 1 \right) \rho_3 \text{Cos} \rho_2 h_j \text{Cot} \rho_3 h_j - \frac{\rho_1^2}{\rho_2^2} \rho_3 \text{Cot} \rho_3 h_j \right] \\
 & - M_{j+1} \left[\rho_1 \text{Cot} \rho_1 h_j - \rho_3 \text{Cot} \rho_3 h_j + \frac{\rho_1 \left(\frac{\rho_3}{\rho_2} \right)^2 - \rho_1 \left(\left(\frac{\rho_3}{\rho_2} \right)^2 - 1 \right) \text{Cos} \rho_2 h_{j+1}}{\text{Sin} \rho_1 h_{j+1}} \right. \\
 & \left. + \frac{\rho_3 \left\{ \text{Cos} \rho_2 h_{j+1} \left(\frac{\rho_1^2}{\rho_2^2} - 1 \right) - \left(\frac{\rho_1^2}{\rho_2^2} \right) \right\}}{\text{Sin} \rho_3 h_{j+1}} \right] \\
 & + M_{j+2} \left\{ \frac{\rho_1}{\text{Sin} \rho_1 h_{j+1}} + \frac{\rho_3}{\text{Sin} \rho_3 h_{j+1}} \right\} = y_j \{ \rho_1^2 \rho_3 \text{Cot} \rho_3 h_j - \rho_1 \rho_3^2 \text{Cot} \rho_1 h_j \} \\
 & + y_{j+1} \left\{ \rho_1 \rho_3^2 \text{Cot} \rho_1 h_j - \rho_1^2 \rho_3 \text{Cot} \rho_3 h_j + \frac{\rho_1 \rho_3^2}{\text{Sin} \rho_1 h_{j+1}} - \frac{\rho_3 \rho_1^2}{\text{Sin} \rho_3 h_{j+1}} \right\} \\
 & + y_{j+2} \left\{ \frac{\rho_3 \rho_1^2}{\text{Sin} \rho_3 h_{j+1}} - \frac{\rho_1 \rho_3^2}{\text{Sin} \rho_1 h_{j+1}} \right\}
 \end{aligned} \tag{3.11}$$

This equation (3.11) can be written as

$$M_j \Omega_j + M_{j+1} \kappa_j + M_{j+2} \eta_j = p_j; \quad j = 0, 1, 2, \dots, n-2 \tag{3.12}$$

where Ω_j , κ_j and η_j are multiplying elements of M_j , M_{j+1} , M_{j+2} and p_j ; $j = 0, 1, \dots, n-2$ reflects the amount in equation (3.11)'s right-hand side.

In order to acquire the system's unique solution, two additional equations as follows must be solved

$$S_0'''(w_1) = S_1'''(w_1) \quad (3.13)$$

and

$$S_{n-2}'''(w_{n-1}) = S_{n-1}'''(w_{n-1}) \quad (3.14)$$

which gives

$$\begin{aligned} & M_0 [Cot\rho_1 h_0 - Cot\rho_1 h_0 Cosp_2 h_0 + Sin\rho_2 h_0 + Cosp_2 h_0 Cot\rho_3 h_0 - Cot\rho_3 h_0] \\ & - M_1 \left[Cot\rho_1 h_0 - Cot\rho_3 h_0 + \frac{Cosp_2 h_1}{Sin\rho_1 h_1} + \frac{\{Cosp_2 h_1 - 1\}}{Sin\rho_3 h_1} \right] \\ & + M_2 \left\{ \frac{1}{Sin\rho_1 h_1} + \frac{1}{Sin\rho_3 h_1} \right\} = 0 \end{aligned} \quad (3.15)$$

Equation (3.15) may be written as

$$M_0 \mathfrak{E}_0 + M_1 \mathfrak{E}_1 + M_2 \mathfrak{E}_2 = 0 \quad (3.16)$$

$$\begin{aligned} & M_{n-2} [Cot\rho_1 h_{n-2} - Cot\rho_1 h_{n-2} Cosp_2 h_{n-2} + Sin\rho_2 h_{n-2} \\ & + Cosp_2 h_{n-2} Cot\rho_3 h_{n-2} - Cot\rho_3 h_{n-2}] \\ & - M_{n-1} \left[Cot\rho_1 h_{n-2} - Cot\rho_3 h_{n-2} + \frac{Cosp_2 h_{n-1}}{Sin\rho_1 h_{n-1}} + \frac{\{Cosp_2 h_{n-1} - 1\}}{Sin\rho_3 h_{n-1}} \right] \\ & + M_{j+2} \left\{ \frac{1}{Sin\rho_1 h_{n-1}} + \frac{1}{Sin\rho_3 h_{n-1}} \right\} = 0 \end{aligned} \quad (3.17)$$

Equation (3.17) may be written as

$$M_{n-2} \mathfrak{E}_{n-2} + M_{n-1} \mathfrak{E}_{n-1} + M_n \mathfrak{E}_n = 0 \quad (3.18)$$

The combination of equations (3.12), (3.16) and (3.18) gives

$$\begin{bmatrix} \mathfrak{E}_0 & \mathfrak{E}_1 & \mathfrak{E}_2 & 0 & \dots & 0 \\ \Omega_0 & \kappa_0 & \eta_0 & 0 & \dots & 0 \\ 0 & \Omega_1 & \kappa_1 & \eta_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \\ 0 & 0 & \dots & \Omega_{n-2} & \kappa_{n-2} & \eta_{n-2} \\ 0 & 0 & \dots & \mathfrak{E}_{n-2} & \mathfrak{E}_{n-1} & \mathfrak{E}_n \end{bmatrix} \begin{bmatrix} M_0 \\ M_1 \\ M_2 \\ M_3 \\ \vdots \\ M_n \end{bmatrix} = \begin{bmatrix} 0 \\ p_0 \\ p_1 \\ p_2 \\ \vdots \\ 0 \end{bmatrix} \quad (3.19)$$

The M_j values found through (3.19) have been applied in equations (i) to (iv) in order to find the values of a_j , b_j , c_j and d_j .

4 The scheme portrayal over the second-order linear BVPs system

This study considers the following type of second-order linear BVPs system

$$\frac{d^2 y}{dx^2} = l(x) \frac{dy}{dx} + m(x) \frac{dz}{dx} + n(x)y(x) + p(x)z(x) + q(x) \quad (4.1)$$

$$\frac{d^2 z}{dx^2} = f(x) \frac{dy}{dx} + g(x) \frac{dz}{dx} + h(x)y(x) + \varnothing(x)z(x) + \varphi(x) \quad (4.2)$$

with

$$y(a) = \vartheta_0, y(b) = \vartheta_n, y(c) = \eta_0, y(d) = \eta_n \quad (4.3)$$

where $l(x), m(x), n(x), p(x), q(x), f(x), g(x), h(x), \varnothing(x)$ and $\varphi(x)$ are continuous functions in the interval $y \in [a, b]$, and $z \in [c, d]$ and $\vartheta_0, \vartheta_n, \eta_0, \eta_n$ are constants.

The vector form of this system can be

$$\begin{pmatrix} y''(x) \\ z''(x) \end{pmatrix} = \begin{pmatrix} l(x) & m(x) \\ f(x) & g(x) \end{pmatrix} \begin{pmatrix} y'(x) \\ z'(x) \end{pmatrix} + \begin{pmatrix} n(x) & p(x) \\ h(x) & \varnothing(x) \end{pmatrix} \begin{pmatrix} y(x) \\ z(x) \end{pmatrix} + \begin{pmatrix} q(x) \\ \varphi(x) \end{pmatrix} \quad (4.4)$$

By applying equation (3.12) on SO-BVPs given in equation (4.4) with $\tilde{M}_j = [y_j'', z_j'']^T$, the corresponding equation is

$$\tilde{M}_j + \tilde{M}_{j+1} \tilde{\kappa}_j + \tilde{M}_{j+2} \tilde{\eta}_j = \tilde{p}_j, j = 0, 1, \dots, n-2 \quad (4.5)$$

where $\tilde{\kappa}_j, \tilde{\eta}_j$ are multiplicative factors of $\tilde{M}_{j+1}, \tilde{M}_{j+2}$ respectively.

When a function ρ is evaluated at the knots, the operator Θ is defined as

$$\Theta \rho_j = \rho(\rho_{j+2} + \rho_{j-2}) + q(\rho_{j+1} + \rho_{j-1}) + r \rho_j$$

this creates the relationships shown below:

$$\Theta \tilde{M}_j = \frac{1}{h_j^2} \left[\ell(u_{j+2} + u_{j-2}) + 2(m - \ell)(u_{j+1} + u_{j-1}) + (2\ell - 4m)u_j \right] \quad (4.6)$$

where

$$u_j = [y_j, z_j]^T$$

$$\rho = \ell_1 + \frac{l}{6}, \bar{\rho} = \ell_1 + \frac{l}{6},$$

$$q = 2 \left[\frac{1}{6} (2\ell + m) - (\ell_1 - m_1) \right],$$

$$r = 2 \left[\frac{1}{6} (\ell + 4m) + (\ell_1 - 2m_1) \right],$$

$$\begin{aligned}\ell &= \left(\frac{1}{\omega^2}\right)(\omega \csc \omega - 1), \\ m &= \left(\frac{1}{\omega^2}\right)(1 - \omega \cot \omega), \\ \ell_1 &= \frac{1}{\omega^2} \left(\frac{1}{6} - \ell\right), m_1 = \frac{1}{\omega^2} \left(\frac{1}{3} - m\right), \\ \omega &= (\rho_1 + \rho_2) h_j.\end{aligned}$$

Using the relation (4.6) in equation (4.4), it results in

$$\begin{aligned}& (-\ell - \rho h_j^2 \tilde{\kappa}_{j-2}) u_{j-2} + (-2(m - \ell) - h_j^2 q \tilde{\eta}_{j-1}) u_{j-1} + (-2\ell + 4m - r h_j^2 \tilde{p}_j) u_j \\& + (-2(m - \ell) - h_j^2 q \tilde{\kappa}_{j+1}) u_{j+1} + (-\ell - \rho h_j^2 \tilde{\eta}_{j+2}) u_{j+2} \\& = -h_j^2 (\rho \tilde{p}_{j-2} + q \tilde{p}_{j-1} + r \tilde{p}_j + q \tilde{p}_{j+1} + \rho \tilde{p}_{j+2}), j = 2(1)n - 2\end{aligned}\quad (4.7)$$

To derive the system's unique solution, two additional equations are needed, and they must take into consideration the boundary conditions listed below:

$$u_1 - 2u_2 + u_3 = \frac{h_j^2}{12} (u_1'' + 10u_2'' + u_3''), j = 1, \quad (4.8)$$

$$u_{n-3} - 2u_{n-2} + u_{n-1} = \frac{h_j^2}{12} (u_{n-3}'' + 10u_{n-2}'' + u_{n-1}''), j = n - 1, \quad (4.9)$$

Using it over equation (4.4)

$$\begin{aligned}& \left(-1 - \frac{h_j^2}{12} \tilde{\kappa}_1\right) u_1 + \left(2 - \frac{10h_j^2}{12} \tilde{\eta}_2\right) u_2 + \left(-1 - \frac{h_j^2}{12} \tilde{p}_3\right) u_3 = -\frac{h_j^2}{12} [\tilde{p}_1 + 10\tilde{p}_2 + \tilde{p}_3], \\& \left(-1 - \frac{h_j^2}{12} \tilde{\kappa}_{n-3}\right) u_{n-3} + \left(2 - \frac{10h_j^2}{12} \tilde{\eta}_{n-2}\right) u_{n-2} + \left(-1 - \frac{h_j^2}{12} \tilde{p}_{n-1}\right) u_{n-1} \\& = -\frac{h_j^2}{12} [\tilde{p}_{n-3} + 10\tilde{p}_{n-2} + \tilde{p}_{n-1}]\end{aligned}\quad (4.10)$$

By applying the Taylor series to expand equation (4.4), the truncation error obtained at w_j will be

$$\begin{aligned}\tilde{T}_j &= \left[\frac{1}{6}(7\ell + m) - (4\rho + q)\right] h_j^4 u_j^{(4)} + \left[\frac{1}{180}(31\ell + m) - \frac{1}{12}(16\rho + q)\right] h_j^6 u_j^{(6)} \\& + \left[\frac{1}{131,040}(1,611\ell + 31m) - \frac{1}{360}(4\rho + q)\right] h_j^8 u_j^{(8)} + O(h^9)\end{aligned}\quad (4.11)$$

For any combination ℓ and m , considering $\ell + m = \frac{1}{2}$.

Aimed at $\ell = \frac{1}{6}, m = \frac{1}{3}$, with $\rho = \frac{1}{120}, q = \frac{26}{120}, r = \frac{66}{120}$, confers $\tilde{T}_j = O(h^6)$.

5 Convergence analysis

The following matrix structure represents the suggested approach to test convergence:

Consider equations (4.7), (4.8) and (4.9) in their matrix form representation as

$$\left. \begin{aligned} \mathbb{L}\mathbb{U} &= \mathbb{P} + \mathbb{T} \\ \mathbb{L}\mathbb{S} &= \mathbb{P} \\ \mathbb{L}\mathbb{E} &= \mathbb{T} \end{aligned} \right\} \quad (5.1)$$

with $\mathbb{U} = [y_j z_j]^T y_j$ or, $\mathbb{S} = S_j, \mathbb{P} = \mathbb{P}_j, \mathbb{T} = \mathbb{T}_j, \mathbb{E} = \mathbb{U} - \mathbb{S}$ be column vectors of dimension n with $\mathbb{E}$ standing for the discretisation error and $\mathbb{L}$ standing for a tridiagonal matrix,

$$\mathbb{L} = \mathbb{L}_0 + \mathbb{R}, \mathbb{R} = \frac{h_j^2}{12} \mathbb{C}\mathbb{K}, \mathbb{K} = \text{diag}(k_j), j = 2, 3, \dots, n-2.$$

Likewise,

$$\mathbb{L}_0 = \begin{bmatrix} 10 & 1 & 1 & & & \\ 12\ell & 12m & 12\ell & & & \\ & 12\ell & 12m & 12\ell & & \\ & & 12\ell & 12m & 12\ell & \\ & & & \ddots & & \\ & & & & 12\ell & 12m & 12\ell \\ & & & & 12\ell & 12m & 12\ell \\ & & & & 1 & 1 & 10 \end{bmatrix}$$

$$\mathbb{P}_j = \begin{cases} \frac{h_j^2}{12} (u_1'' + 10u_2'' + u_3''), & j = 1, \\ -h_j^2 (\rho \tilde{p}_{j-2} + q \tilde{p}_{j-1} + r \tilde{p}_j + q \tilde{p}_j + q \tilde{p} + \rho \tilde{p}_{j+2}), & j = 2(1)n-2, \\ \frac{h_j^2}{12} (u_{n-3}'' + 10u_{n-2}'' + u_{n-1}''), & j = n-1, \end{cases} \quad (5.2)$$

The idea is to redefine equation $\mathbb{L}\mathbb{E} = \mathbb{T}$ in order to establish a bound on $\|\mathbb{E}\|$ as

$$\begin{aligned} \mathbb{E} &= \mathbb{L}^{-1}\mathbb{T} = (\mathbb{L}_0 + \mathbb{R})^{-1} \mathbb{T} = (\mathbb{I} + \mathbb{L}_0^{-1}\mathbb{R})^{-1} \mathbb{L}_0^{-1}\mathbb{T}, \\ \|\mathbb{E}\| &\leq \|(\mathbb{I} - \mathbb{L}_0^{-1}\mathbb{R})\| \cdot \|\mathbb{L}_0^{-1}\| \cdot \|\mathbb{T}\|, \end{aligned} \quad (5.3)$$

Provided $\|(\mathbb{I} + \mathbb{L}_0^{-1}\mathbb{R})^{-1}\| \leq \|(\mathbb{I} - \mathbb{L}_0^{-1}\mathbb{R})^{-1}\|$ such that in the matrix vector $\|\cdot\|$ defines for the ∞ -norm.

It describes

$$\|\mathbb{E}\| \leq \frac{\|\mathbb{L}_0^{-1}\| \cdot \|\mathbb{T}\|}{1 - \|\mathbb{L}_0^{-1}\| \cdot \|\mathbb{R}\|}, \quad (5.4)$$

provided $\|\mathbb{L}_0^{-1}\| \cdot \|\mathbb{R}\| < 1$.

As per definition Amodio and Sugra (2005),

$$\|\mathbb{L}_0^{-1}\| \leq \frac{1}{8h^2} ((b-a)^2 + h_j^2) = O(h_j^{-2}) \quad (5.5)$$

By equation (4.11)

$$\|\mathbb{T}\| = h_j^4 \left[\frac{1}{6} (7\ell + m) + (4p + q) \right] u_j^{(4)} \quad (5.6)$$

Using equations (5.4)–(5.6) it is noted that

$$\|\mathbb{E}\| \leq \frac{\|\mathbb{L}_0^{-1}\| \cdot \|\mathbb{T}\|}{1 - \|\mathbb{L}_0^{-1}\| \cdot \left\| \frac{h_j^2}{12} \mathbb{C}\mathbb{K} \right\|} \cong O(h_j^2), \quad (5.7)$$

The equation (5.7) approves that the proposed approach is second order convergent.

6 Numerical illustrations

To show the effectiveness of the strategy, the proposed scheme is applied to three systems of BVPs selected from Shahna and Khan (2021). To test the effectiveness of the suggested system in each challenge, the step size is chosen at random, and it is discovered to be well coordinated.

Problem 1: Taking into account the following BVP

$$\left. \begin{aligned} \frac{d^2 y}{dx^2} + (2x-1) \frac{dy}{dx} + \cos(\pi x) \frac{dz}{dx} &= g_1(x) \\ \frac{d^2 z}{dx^2} + xy &= g_2(x) \end{aligned} \right\}$$

where

$$g_1(x) = -\pi^2 \sin(\pi x) + (2x-1)\pi \cos(\pi x) + (2x-1)\cos(\pi x),$$

$$g_2(x) = 2 + x \sin(\pi x),$$

with $y(0) = 0$, $y(1) = 0$, $z(0) = 0$, $z(1) = 0$, $0 < x < 1$. The exact solutions are $y(x) = \sin(\pi x)$ and $z(x) = x^2 - x$.

Solution: The values of y and z that correspond to the various values of x chosen at different step sizes are shown in Table 1 along with the precise solutions. The approximate solutions are discovered to be extremely near the precise solutions.

Table 1 Numerical results of Problem 1 using proposed method

x	h_i	$y(x)$ (exact solution)	y_i (approximated) (proposed method)	$z(x)$ (exact solution)	z_i (approximated) (proposed method)	CPU time (in seconds)
0.275000000000000000	0.275000000000000000	0.76040596560003100	0.760405965572547000	-0.199375000000000000	-0.199374999999837000	1.312
0.400000000000000000	0.125000000000000000	0.95105651629515400	0.951056516249521000	-0.240000000000000000	-0.2399999999999407000	1.446
0.437500000000000000	0.037500000000000000	0.98078528040323000	0.980785280327016000	-0.246093750000000000	-0.246093749999164000	1.54
0.468750000000000000	0.031250000000000000	0.99518472667219700	0.995184726415218000	-0.249023437500000000	-0.249023437498451000	1.607
0.474375000000000000	0.005624999999999999	0.99676135804373100	0.996761357668893000	-0.249343359375000000	-0.249343359368505000	1.655
0.500000000000000000	0.025625000000000000	1.000000000000000000	0.999999999413654000	-0.250000000000000000	-0.249999999990505000	1.776
0.568541666666666700	0.068541666666666690	0.97690593711580200	0.976905934746018000	-0.245302039930555000	-0.245302039905087000	1.863
0.60880952380952400	0.04026785714285710	0.94214106256654500	0.942141058331848000	-0.238160487528345000	-0.23816048748297000	1.939
0.64907738095238100	0.04026785714285700	0.89231866794479000	0.892318661355159000	-0.227775934488379000	-0.227775934422481000	2.011
0.68934523809523800	0.04026785714285700	0.82823502648061300	0.828235017694963000	-0.214148380810658000	-0.214148380724760000	2.066
0.72961309523809500	0.04026785714285710	0.75091433800565800	0.750914320435834000	-0.197277826495182000	-0.197277826168638000	2.127
0.76988095238095300	0.04026785714285800	0.66159235995051300	0.661592324271577000	-0.177164271541950000	-0.177164270956052000	2.209
0.81014880952381000	0.04026785714285700	0.56169665717878900	0.561696591691167000	-0.153807715950963000	-0.153807715153501000	2.286
0.850416666666666700	0.04026785714285710	0.45282378630966200	0.452823699760938000	-0.127208159722222000	-0.127208158800743000	2.399
0.89068452380952400	0.04026785714285700	0.33671377917445800	0.336713213687244000	-0.097365602855725500	-0.097365598286976200	2.471
0.93095238095238200	0.04026785714285800	0.21522323321861900	0.215221588094970000	-0.064280045351473000	-0.064280038766678500	2.584
1.000000000000000000	0.06904761904761800	0.00000000000000012	0.000000963462662399	0.000000000000000000	-0.00000008246587955	2.693

Table 2 Numerical results of Problem 2 using proposed method

x	h_j	$y(x)$ (exact solution)	y_i (approximated) (proposed method)	$z(x)$ (exact solution)	z_i (approximated) (proposed method)	CPU time (in seconds)
0.2750000000000000	0.2750000000000000	-0.1993750000000000	-0.1993750000000000	0.1993750000000000	0.1993750000000000	1.416
0.4000000000000000	0.1250000000000000	-0.2400000000000000	-0.2400000000000000	0.2400000000000000	0.2400000000000000	1.551
0.4375000000000000	0.0375000000000000	-0.2460937500000000	-0.2460937500000000	0.2460937500000000	0.2460937500000000	1.637
0.4687500000000000	0.0312500000000000	-0.2490234375000000	-0.2490234375000000	0.2490234375000000	0.2490234375000000	1.717
0.4743750000000000	0.0056249999999999	-0.2493433593750000	-0.2493433593750000	0.2493433593750000	0.2493433593750000	1.789
0.5000000000000000	0.0256250000000000	-0.2500000000000000	-0.2500000000000000	0.2500000000000000	0.2500000000000000	1.882
0.5685416666666670	0.0685416666666690	-0.2453020399305560	-0.2453020399305560	0.2453020399305560	0.2453020399305560	1.957
0.6088095238095240	0.0402678571428571	-0.2381604875283450	-0.2381604875283450	0.2381604875283450	0.2381604875283450	2.053
0.6490773809523810	0.0402678571428570	-0.2277759344883790	-0.2277759344883790	0.2277759344883790	0.2277759344883790	2.127
0.6893452380952380	0.0402678571428570	-0.2141483808106580	-0.2141483808106580	0.2141483808106580	0.2141483808106580	2.182
0.7296130952380950	0.0402678571428571	-0.1972778264951810	-0.1972778264951820	0.1972778264951810	0.1972778264951780	2.231
0.7698809523809530	0.0402678571428580	-0.1771642715419500	-0.1771642715419500	0.1771642715419500	0.1771642715419440	2.313
0.8101488095238100	0.0402678571428570	-0.1538077159509630	-0.1538077159509640	0.1538077159509630	0.1538077159509550	2.398
0.8504166666666670	0.0402678571428571	-0.1272081597222200	-0.1272081597222300	0.1272081597222200	0.12720815972221300	2.517
0.8906845238095240	0.0402678571428570	-0.0973656028557255	-0.09736560285572930	0.0973656028557255	0.09736560285567980	2.591
0.9309523809523820	0.0402678571428580	-0.0642800453514730	-0.06428004535147860	0.06428004535147300	0.06428004535140720	2.697
1.0000000000000000	0.0690476190476180	0.0000000000000000	-0.0000000000000743	0.0000000000000000	-0.00000000000008247	2.818

Table 3 Numerical results of Problem 3 using proposed method

x	h_j	$y(x)$ (exact solution)	$\bar{y}_i$ (approximated) (proposed method)	$z(x)$ (exact solution)	$\bar{z}_i$ (approximated) (proposed method)	CPU time (in seconds)
0.275000000000000000	0.275000000000000000	-0.19687152929318200	-0.196871529294729000	0.199375000000000000	0.1993750000004632000	1.164
0.400000000000000000	0.125000000000000000	-0.23365100538519000	-0.233651005388445000	0.240000000000000000	0.2400000000006933000	1.317
0.437500000000000000	0.037500000000000000	-0.23831789467721500	-0.238317894683837000	0.246093750000000000	0.2460937500008357000	1.381
0.468750000000000000	0.031250000000000000	-0.24000359422995700	-0.240003594264534000	0.249023437500000000	0.249023437531246000	1.466
0.484375000000000000	0.015625000000000000	-0.24010353808297300	-0.240103538139397000	0.249755859375000000	0.249755859429949000	1.525
0.500000000000000000	0.015625000000000000	-0.23971276930210200	-0.239712769377558000	0.250000000000000000	0.250000000074949000	1.618
0.568541666666666700	0.068541666666666690	-0.23229876439412500	-0.232298764518693000	0.245302039930555000	0.245302040185243000	1.712
0.60880952380952400	0.04026785714285710	-0.22371842330158100	-0.223718423627038000	0.238160487528345000	0.238160487952903000	1.795
0.64907738095238100	0.04026785714285700	-0.21211576178675600	-0.212115762340970000	0.227775934488379000	0.227775935139624000	1.863
0.68934523809523800	0.04026785714285700	-0.19758638918897200	-0.197586389943203000	0.214148380810658000	0.214148381669632000	1.925
0.72961309523809500	0.04026785714285710	-0.18023484733353100	-0.180234849456987000	0.197277826495182000	0.197277829760594000	1.991
0.76988095238095300	0.04026785714285800	-0.16017430973954100	-0.160174313970999000	0.177164271541955000	0.177164277400926000	2.065
0.81014880952381000	0.04026785714285700	-0.13752626025715900	-0.137526266488617000	0.153807715950963000	0.153807723925585000	2.138
0.850416666666666700	0.04026785714285710	-0.11242015187879800	-0.112420161420966000	0.127208159722222000	0.127208168937012000	2.241
0.89068452380952400	0.04026785714285700	-0.08499304651131220	-0.084993089159168700	0.097365602855725500	0.097365644101482100	2.333
0.93095238095238200	0.04026785714285800	-0.05538923653693260	-0.055389301994778400	0.064280045351473000	0.064280107666038900	2.417
1.000000000000000000	0.06904761904761800	0.000000000000000000	0.000000083126457635	0.000000000000000000	-0.00000092465879549	2.508

Problem 2: Consider the system of BVPs

$$\left. \begin{aligned} \frac{d^2 y}{dx^2} + x \frac{dy}{dx} + x \frac{dz}{dx} &= 2 \\ \frac{d^2 z}{dx^2} + 2x \frac{dy}{dx} + 2x \frac{dz}{dx} &= -2 \end{aligned} \right\}$$

with $y(0) = 0$, $y(1) = 0$, $z(0) = 0$, $z(1) = 0$, $0 < x < 1$. The exact solutions are $y(x) = x^2 - x$ and $z(x) = x - x^2$.

Solution: Table 2 displays the various x values along with random step sizes h_j , together with the obtained and precise y solutions. The estimated solutions are discovered to be extremely near to the precise solutions.

Problem 3: Taking into account the nonlinear system of BVPs

$$\left. \begin{aligned} \frac{d^2 y}{dx^2} + x \frac{dy}{dx} + \cos(\pi x) \frac{dz}{dx} &= g_1(x) \\ \frac{d^2 z}{dx^2} + x \frac{dy}{dx} + xy^2 &= g_2(x) \end{aligned} \right\}$$

where

$$g_1(x) = \sin(x) + (x^2 - x + 2)\cos(x) + (1 - 2x)\cos(\pi x),$$

$$g_2(x) = x(x-1)^2 + x\sin(x) + \sin^2(x) + (x^2 - x)\cos(x) - 2,$$

with $y(0) = 0$, $y(1) = 0$, $z(0) = 0$, $z(1) = 0$, $0 < x < 1$. The exact solutions are $y(x) = (x - 1)\sin(x)$ and $z(x) = x - x^2$.

Solution: Table 3 displays the various x values along with random step sizes h_j , together with the obtained and precise y solutions. The estimated solutions are discovered to be extremely near to the precise solutions.

7 Comparative study and discussion

The addition of the triplet parameters (ρ_1, ρ_2, ρ_3) to the non-polynomial spline's trigonometric section increases the scheme's accuracy when applied to a broad system of second order differential equations. They are superior to non-polynomial single parameter spline methods, B-spline approaches, and variational iterative approaches because they provide a continuous approximation to the function and its higher derivatives. Results of the proposed method are compared to those of B-spline, second order cubic non-polynomial single parameter spline, fourth order cubic non-polynomial single parameter spline, and variational iteration method using maximum absolute errors (MAEs) in tabular and graphical form. Comparative analysis of Example 1 is shown in Table 4, while Examples 2 and 3 are shown in Tables 5 and 6.

Table 4 Comparison of MAE by proposed method and other existing methods for Problem 1

<i>Method</i>	<i>Maximum absolute error</i>
$y(x)$ error by proposed method	2.748378E-11
$y(x)$ error by fourth order method (Shahna and Khan, 2021)	1.2207E-08
$y(x)$ error by second order method (Shahna and Khan, 2021)	9.3019E-05
$z(x)$ error by proposed method	1.63215487E-13
$z(x)$ error by for fourth order method (Shahna and Khan, 2021)	6.4875E-10
$z(x)$ error by second order method (Shahna and Khan, 2021)	4.6494E-06

Table 5 Comparison of MAE by proposed method and other existing methods for Problem 2

<i>Method</i>	<i>Maximum absolute error</i>
$y(x)$ error by proposed method	5.2155E-19
$y(x)$ error by fourth order method (Shahna and Khan, 2021)	8.3267E-17
$y(x)$ error by fourth order method (Shahna and Khan, 2021)	7.4940E-16
$y(x)$ error by B-spline method (Caglar and Caglar, 2009)	3.4139E-15
$z(x)$ error by proposed method	1.6322E-18
$z(x)$ error by fourth order method (Shahna and Khan, 2021)	2.4980E-16
$z(x)$ error by second order method (Shahna and Khan, 2021)	3.8858 E-16
$z(x)$ error by B-spline method (Caglar and Caglar, 2009)	1.8457E-15

Table 6 Comparison of MAE by proposed method and other existing methods for Problem 3

<i>Method</i>	<i>Maximum absolute error</i>
$y(x)$ error by proposed method	1.5470E-12
$y(x)$ error by fourth order method (Shahna and Khan, 2021)	1.4866E-09
$y(x)$ error by second order method (Shahna and Khan, 2021)	9.1538E-06
$y(x)$ error by variational iteration method (Lu, 2007)	4.5900E-02
$z(x)$ error by proposed method	4.6322E-12
$z(x)$ error by fourth-order method (Shahna and Khan, 2021)	2.8044E-09
$z(x)$ error by second-order method (Shahna and Khan, 2021)	3.2445E-06

In addition, Figures 1 to 3 show a graphic comparison of the proposed scheme with the aforementioned three approaches to solve Problems 1, 2, and 3. This comparison reveals unequivocally that the proposed technique is superior to these three widely used approaches.

8 Established theorem on proposed numerical method and its real word applications

The main result of this study establishes the accuracy, stability, and efficiency of the proposed numerical method for solving SO-BVPs, as stated in the following theorem.

Figure 1 Comparison of MAE by second and fourth order one parameter non-polynomial spline and proposed method in Problem 1 (see online version for colours)

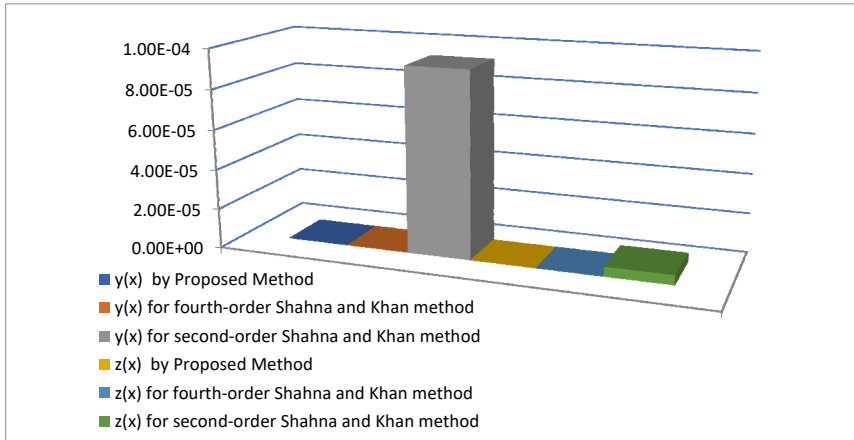
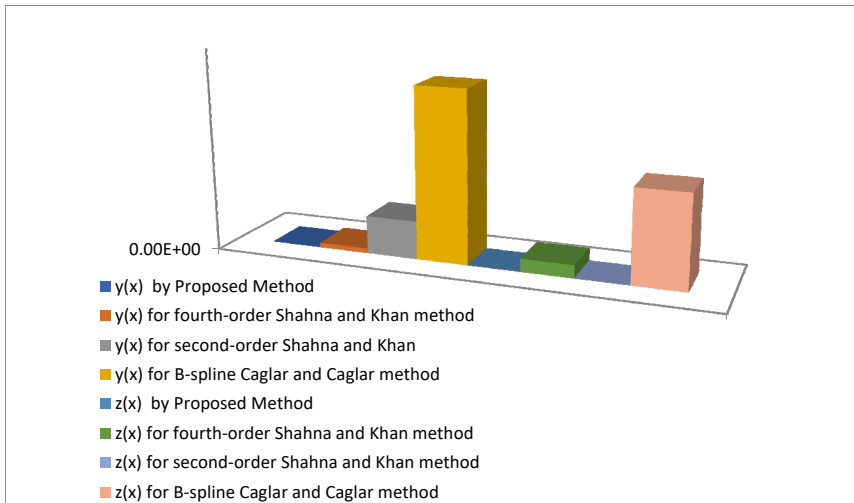


Figure 2 Comparison of MAE by second and fourth order one parameter non-polynomial spline, B-spline and proposed method in Problem 2 (see online version for colours)



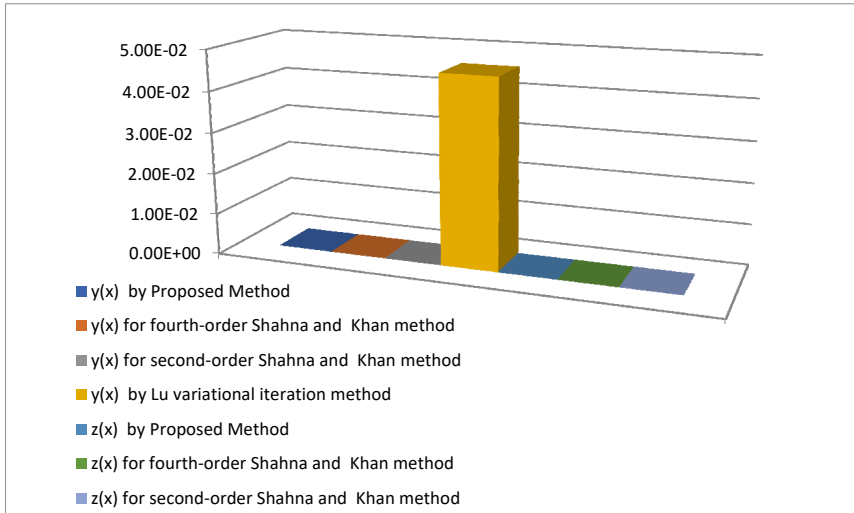
8.1 Theorem (convergence and accuracy of the proposed numerical method)

Consider a system of SO-BVPs with given boundary conditions (Srivastava, 2021). The proposed numerical method, based on cubic splines with non-polynomial triplet parameters, provides an approximate solution over a defined interval with a chosen step size. Under appropriate smoothness conditions on the given function and its solution, the proposed method satisfies the following properties:

- 1 Convergence: The method achieves fourth-order accuracy, meaning the error reduces significantly as the step size decreases.

- 2 Stability: The numerical scheme remains stable for different step sizes due to the additional parameters incorporated into the cubic spline formulation.
- 3 Computational efficiency: Compared to traditional spline, collocation, and finite difference methods, the proposed scheme provides better accuracy while reducing computational effort.

Figure 3 Comparison of MAE by second and fourth order one parameter non-polynomial spline, variational iteration and proposed methods in Problem 3 (see online version for colours)



8.2 Real word applications

The proposed numerical method's effectiveness in solving SO-BVPs makes it a valuable tool for various real-world applications. Here are some real-world applications where the proposed method for solving SO-BVPs could be effectively applied:

- 1 Structural engineering: Modelling and analysing beam deflection under various loading conditions in mechanical and civil engineering.
- 2 Fluid dynamics: Solving problems related to laminar flow in pipes or channels, where differential equations describe velocity and pressure distribution.
- 3 Thermal analysis: Modelling heat conduction in materials with varying properties, crucial in mechanical and aerospace engineering.
- 4 Environmental modelling: Simulating pollution dispersion or temperature distribution in atmospheric or aquatic systems.
- 5 Electrical engineering: Solving differential equations for circuit analysis, particularly in systems involving inductance and capacitance.

This versatility highlights the proposed method's potential to improve numerical solutions across diverse engineering and scientific applications with greater accuracy.

9 Conclusions

The qualities of non-polynomial splines are used in this work to improve its features by employing three more parameters. Because of the trigonometric features, the existence of triplet parameters in the trigonometric component of the spline has made it more efficient. Because of its application in unequal step size, the proposed study becomes more useful. Other conventional approaches, such as B-spline, one parameter non-polynomial spline, and variational iteration, are only applicable to problems with equal step size. Due to the suggested system's applicability for unequal step size, obtaining the ordinate values at a given abscissa is thus a simple task that is time-consuming for existing approaches. The cherry on top is that this research has also been applied to a broader system of second order differential equations. The effectiveness of the strategy has been proved by the convergence analysis. Understanding SO-BVPs is crucial for modelling real-world phenomena. This study proposes a novel numerical method using cubic splines and non-polynomial triplet parameters, enhancing accuracy and adaptability across different step sizes. Future research can explore extending this method to higher-order BVPs and nonlinear systems, improving accuracy and stability. Adaptive step-size strategies and hybrid approaches integrating machine learning could enhance efficiency. Investigating error bounds and theoretical guarantees will strengthen reliability. Applications in fluid dynamics, biomechanics, and quantum mechanics offer exciting directions.

Declarations

The author has no conflict of interests, as it is a single author manuscript.

References

- Abu-Arquab, O., Abo-Hammour, Z. and Momani, S. (2014) 'Application of continuous genetic algorithm for nonlinear system of second-order boundary value problems', *Applied Mathematics & Information Sciences*, Vol. 8, No. 1, p.235.
- Achchab, B., Bouihat, K. and El-Bouayadi, A. (2023) 'Adaptive finite element method for Wick stochastic partial differential equations', *International Journal of Mathematical Models and Numerical Optimization*, Vol. 13, No. 4, pp.400–415.
- Adeyanju, A.A. (2023) 'Stability and boundedness criteria of solutions of a certain system of second order differential equations', *Annali dell'Università di Ferrara*, Vol. 69, No. 1, pp.81–93.
- Ala'yed, O. et al. (2023) 'Cubic B-Spline method for the solution of the quadratic Riccati differential equation', *AIMS Mathematics*, Vol. 8, No. 4, pp.9576–9584.
- Al-Said, E.A. (2001) 'The use of cubic splines in the numerical solution of a system of second-order boundary value problems', *Computers & Mathematics with Applications*, Vol. 42, Nos. 6–7, pp.861–869.
- Amodio, P. and Sgura, I. (2005) 'High-order finite difference schemes for the solution of second-order BVPs', *Journal of Computational and Applied Mathematics*, April, Vol. 176, No. 1, pp.59–76, DOI: 10.1016/j.cam.2004.07.008.
- Arquab, O.A., Al-Smadi, M., Momani, S. and Hayat, T. (2017) 'Application of reproducing kernel algorithm for solving second-order, two-point fuzzy boundary value problems', *Soft Computing*, Vol. 21, No. 23, pp.7191–7206.

- Arumugam, P. et al. (2025) 'A quintic spline-based computational method for solving singularly perturbed periodic boundary value problems', *Axioms*, Vol. 14, No. 1, p.73.
- Babu, A. and Asharaf, N. (2023) 'Numerical solution of nonlinear sine-Gordon equation using modified cubic B-spline-based differential quadrature method', *Computational Methods in Differential Equations*, Vol. 11, No. 2, pp.369–386.
- Bataineh, A.S., Noorani, M.S.M. and Hashim, I. (2009) 'Modified homotopy analysis method for solving systems of second-order BVPs', *Communications in Nonlinear Science and Numerical Simulation*, Vol. 14, No. 2, pp.430–442.
- Cabada Fernández, A., Enguiça, R. and López Somoza, L. (2017) *Positive Solutions for Second-Order Boundary-Value Problems with Sign Changing Green's Functions*, arXiv preprint arXiv:1706.07395.
- Caglar, N. and Caglar, H. (2009) 'B-spline method for solving linear system of second-order boundary value problems', *Computers & Mathematics with Applications*, Vol. 57, No. 5, pp.757–762.
- Cheng, X. and Zhong, C. (2005) 'Existence of positive solutions for a second-order ordinary differential system', *Journal of Mathematical Analysis and Applications*, Vol. 312, No. 1, pp.14–23.
- Daba, I.T. et al. (2025) 'Numerical investigation of singularly perturbed time lag parabolic differential-difference equations', *Heliyon*, Vol. 11, No. 1, p.e41215.
- Das, A., Govindarao, L. and Mohapatra, J. (2022) 'A second order weighted monotone numerical scheme for time-delayed parabolic initial-boundary-value problem involving a small parameter', *International Journal of Mathematical Models and Numerical Optimization*, Vol. 12, No. 3, pp.250–265.
- Dehghan, M. and Saadatmandi, A. (2007) 'The numerical solution of a nonlinear system of second-order boundary value problems using the sinc-collocation method', *Mathematical and Computer Modelling*, Vol. 46, Nos. 11–12, pp.1434–1441.
- El-Gamel, M. (2012) 'Sinc-collocation method for solving linear and nonlinear system of second-order boundary value problems', *Applied Mathematics*, Vol. 3, No. 11, pp.1627–1633.
- Erdogan, F. (2024) 'A second order numerical method for singularly perturbed Volterra integro-differential equations with delay', *International Journal of Mathematical and Computational Engineering*, Vol. 2, No. 1, pp.85–96.
- Geng, F. and Cui, M. (2007) 'Solving a nonlinear system of second order boundary value problems', *Journal of Mathematical Analysis and Applications*, Vol. 327, No. 2, pp.1167–1181.
- Güngör, G. and Altun, I. (2024) 'An existence result for second-order boundary-value problems via new fixed-point theorems on quasi-metric space', *Symmetry*, Vol. 16, No. 1, p.99.
- Hariharan, G., Kavitha, J., Seethalakshmi, R. and Jafari, H. (2025) 'An efficient approximation method for nonlinear singular value problems arising in astrophysics: an operational matrix approach', *Astrophysics and Space Science*, Vol. 370, Art, No. 9, pp.1–10.
- Jena, S.R. and Gebremedhin, G.S. (2023) 'Decatic B-spline collocation scheme for approximate solution of burgers' equation', *Numerical Methods for Partial Differential Equations*, Vol. 39, No. 3, pp.1851–1869.
- Karthick, S. and Subburayan, V. (2022) 'Finite difference methods with linear interpolation for solving a coupled system of hyperbolic delay differential equations', *International Journal of Mathematical Models and Numerical Optimization*, Vol. 12, No. 4, pp.365–380.
- Kumar, M. and Srivastava, P.K. (2008) 'Computational techniques for solving differential equations by quadratic, quartic and octic spline', *Advances in Engineering Software*, Vol. 39, No. 8, pp.646–653.
- Kumar, M. and Srivastava, P.K. (2009) 'Computational techniques for solving differential equations by cubic, quintic, and sextic spline', *International Journal for Computational Methods in Engineering Science and Mechanics*, February, Vol. 10, No. 1, pp.108–115, DOI: 10.1080/15502280802623297.

- Lang, F-G. and Xu, X-P. (2012) 'Quintic B-spline collocation method for second order mixed boundary value problem', *Computer Physics Communications*, Vol. 183, No. 4, pp.913–921.
- Liu, L-B., Liu, H-W. and Chen, Y. (2011) 'Polynomial spline approach for solving second-order boundary-value problems with Neumann conditions', *Applied Mathematics and Computation*, Vol. 217, No. 16, pp.6872–6882.
- Lu, J. (2007) 'Variational iteration method for solving a nonlinear system of second-order boundary value problems', *Computers & Mathematics with Applications*, Vol. 54, Nos. 7–8, pp.1133–1138.
- Lubo, G.T. and Duressa, v (2023) 'Linear B-spline finite element method for solving delay reaction diffusion equation', *Computational Methods in Differential Equations*, Vol. 11, No. 1, pp.161–174.
- Mangal, S. and Gupta, V. (2025) 'Exponential B-spline collocation method with Richardson extrapolation for Black-Scholes equation', *Journal of Applied Mathematics and Computation*, available online, pp.1–31, DOI: 10.1007/s12190-025-02377-4.
- Masti, I. and Sayevand, K. (2024) 'On collocation-Galerkin method and fractional B-spline functions for a class of stochastic fractional integro-differential equations', *Mathematics and Computation in Simulation*, Vol. 216, No. 1, pp.263–287.
- Motsa, S.S., Sibanda, P. and Shateyi, S. (2010) 'A new spectral-homotopy analysis method for solving a nonlinear second order BVP', *Communications in Nonlinear Science and Numerical Simulation*, September, Vol. 15, No. 9, pp.2293–2302, DOI: 10.1016/j.cnsns.2009.09.019.
- Negero, N.T. (2023) 'Fitted cubic spline in tension difference scheme for two-parameter singularly perturbed delay parabolic partial differential equations', *Partial Differential Equations and Applications in Mathematics*, Vol. 8, No. 1, p.100530.
- Palav, M.S. and Pradhan, V.H. (2025) 'Fourth-order hyperbolic polynomial B-splines for advection-diffusion equation', *Applied Mathematics and Computation*, Vol. 484, No. 1, p.128992.
- Parumasur, N., Adetona, R.A. and Singh, P. (2023) 'Efficient solution of burgers', modified burgers' and KdV-burgers' equations using B-spline approximation functions', *Mathematics*, Vol. 11, No. 8, p.1847.
- Ramadan, M.A., Lashien, I.F. and Zahra, W.K. (2007) 'Polynomial and nonpolynomial spline approaches to the numerical solution of second order boundary value problems', *Applied Mathematics and Computation*, Vol. 184, No. 2, pp.476–484.
- Ramos, H. and Rufai, M.A. (2019) 'A third-derivative two-step block Falkner-type method for solving general second-order boundary-value systems', *Mathematics and Computers in Simulation*, Vol. 165, No. 1, pp.139–155.
- Rawat, A.K., Deep, G., Dhiman, N. and Chauhan, A. (2023) 'Convergence analysis and an efficient numerical technique for the solution of Benjamin Bona Mahony partial differential equation', *International Journal of Mathematical Models and Numerical Optimization*, Vol. 13, No. 2, pp.150–165.
- Sabir, Z. et al. (2024) 'Integrated neuro-evolution heuristic with sequential quadratic programming for second-order prediction differential models', *Numerical Methods for Partial Differential Equations*, Vol. 40, No. 1, p.e22692.
- Sahihi, H., Allahviranloo, T. and Abbasbandy, S. (2020) 'Solving system of second-order BVPs using a new algorithm based on reproducing kernel Hilbert space', *Applied Numerical Mathematics*, May, Vol. 151, pp.27–39, DOI: 10.1016/j.apnum.2019.12.008.
- Shahna and Khan, A. (2021) 'A new algorithm for solving generalized systems of second-order boundary value problems using nonpolynomial spline technique', *Proceedings of the National Academy of Sciences, India Section A: Physical Sciences*, Vol. 91, pp.225–235.
- Singh, P., Srivastava, P.K., Patney, R.K., Joshi, S.D. and Saha, K. (2013) 'Nonpolynomial spline based empirical mode decomposition', in *2013 International Conference on Signal Processing and Communication (ICSC)*, IEEE, pp.435–440.

- Srivastava, P.K. (2014) 'Study of differential equations with their polynomial and nonpolynomial spline based approximation', *Acta Technica Corviniensis-Bulletin of Engineering*, Vol. 7, No. 3, p.139.
- Srivastava, P.K. (2015) 'Application of higher order splines for boundary value problems', *World Academy of Science, Engineering and Technology, International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering*, Vol. 9, No. 2, pp.115–121.
- Srivastava, P.K. (2019) 'A spline-based computational technique applicable for solution of boundary value problem arising in human physiology', *International Journal of Computing Science and Mathematics*, Vol. 10, No. 1, pp.46–57.
- Srivastava, P.K. (2021) 'Nonpolynomial twin parameter spline approach to treat boundary-value problems arising in engineering problems', *Computational and Applied Mathematics*, Vol. 40, Art. No. 69, pp.1–18.
- Srivastava, P.K. and Kumar, M. (2011) 'Numerical treatment of nonlinear third order boundary value problem', *Applied Mathematics*, Vol. 2, No. 8, p.959.
- Srivastava, P.K. and Kumar, M. (2012) 'Numerical algorithm based on quintic nonpolynomial spline for solving third-order boundary value problems associated with draining and coating flows', *Chinese Annals of Mathematics, Series B*, Vol. 33, No. 6, pp.831–840.
- Srivastava, P.K., Kumar, M. and Mohapatra, R.N. (2011) 'Quintic nonpolynomial spline method for the solution of a second-order boundary-value problem with engineering applications', *Computers & Mathematics with Applications*, Vol. 62, No. 4, pp.1707–1714.
- Srivastava, P.K., Kumar, M. and Mohapatra, R.N. (2012) 'Solution of fourth order boundary value problems by numerical algorithms based on nonpolynomial quintic splines', *Journal of Numerical Mathematics and Stochastics*, Vol. 4, No. 1, pp.13–25.
- Srivastav, N. and Barnwal, A.K. (2023) 'Numerical solution of Lane-Emden pantograph delay differential equation: stability and convergence analysis', *International Journal of Mathematical Models and Numerical Optimization*, Vol. 13, No. 1, pp.64–83.
- Sun, H., Mei, L. and Lin, Y. (2021) 'A new algorithm based on improved legendre orthonormal basis for solving second-order BVPs', *Applied Mathematics Letters*, February, Vol. 112, p.106732, DOI: 10.1016/j.aml.2020.106732.
- Thompson, H.B. and Tisdell, C. (2000) 'Systems of difference equations associated with boundary value problems for second order systems of ordinary differential equations', *Journal of Mathematical Analysis and Applications*, Vol. 248, No. 2, pp.333–347.
- Zhang, Y. (2023) 'On the oscillation of certain second-order linear differential equations', *Proceedings of the Royal Society of Edinburgh: Section A Mathematics*, Vol. 153, No. 6, pp.1965–1992.
- Zheng, Y., Lin, Y. and Shen, Y. (2020) 'A new multiscale algorithm for solving second order boundary value problems', *Applied Numerical Mathematics*, Vol. 156, No. 1, pp.528–541.