Check for updates

Mathematical Methods in the Applied Sciences

WILEY

RESEARCH ARTICLE

A Versatile Spectral Matrix Collocation Method for Tackling Bratu, Riccati, and Higher Order Linear and Nonlinear Differential Equations Containing Delay Variables

Nurcan Baykuş Savaşaneril[1] | Mehmet Çevik[2]

[1]Dokuz Eylül University, İzmir Vocational School, Türkiye | [2]İzmir Katip Çelebi University, Department of Mechanical Engineering, Türkiye

Correspondence: Nurcan Baykuş Savaşaneril (nurcan.savasaneril@deu.edu.tr)

Received: 27 May 2025 | **Revised:** 26 July 2025 | **Accepted:** 12 August 2025

Funding: The authors received no specific funding for this work.

Keywords: Bratu | collocation points | linear delay differential equations and nonlinear delay differential equations | Lucas polynomials | residual error analysis | Riccati

ABSTRACT

This study presents a numerical matrix collocation method for solving Bratu equations, Riccati equations, and high-order linear/nonlinear differential equations with delay variables—all of which hold significant importance in the literature. Addressing these widely studied problems from a novel perspective, we develop a new technique that transforms each problem into an algebraic matrix equation based on Lucas polynomials. The proposed method constructs approximate solutions using Lucas polynomials with conventional ordering points, which form the foundational basis of our approach. For each problem type, we explore different solution scenarios and provide tailored computational strategies. Numerical results, presented in tables and figures, demonstrate consistency and effectiveness by comparing our method with existing approaches for various ranking points.

MSC2020 Classification: 33C45, 65D32, 65F10, 34A34, 35J60, 34K11, 34K35

1 | Introduction

Bratu problems, Riccati equations, and higher order linear and nonlinear differential equations containing delay variables play a pivotal role in a wide range of fields, including mathematics, engineering, and applied sciences. These equations are crucial in modeling dynamic systems, control theory, and optimization problems, often arising in disciplines such as fluid dynamics, thermodynamics, and financial mathematics. Their ability to describe complex real-world phenomena, such as heat conduction, population dynamics, and electrical circuits with time delays, makes them essential tools for both theoretical research and practical applications in various industries.

The Bratu problem, a nonlinear differential equation, arises in modeling various physical systems—particularly in heat transfer and fluid dynamics—where it describes an equilibrium state between diffusion and nonlinear reaction terms. Due to its mathematical complexity and practical relevance, this problem has become a key area of research, prompting the development of diverse analytical and numerical solution methods. Extensive studies, such as those documented in [1–18], have explored perturbation techniques, spectral methods, and iterative algorithms to investigate the Bratu equation and derive its solutions. Our approach leverages Lucas polynomials, capitalizing on their superior spectral properties to handle boundary conditions and nonlinear delay terms efficiently. An extensive review of spectral collocation methods can be found in [19]. Furthermore,

© 2025 John Wiley & Sons Ltd.

alternative analytic approaches, such as the homotopy analysis method, have also been proposed for nonlinear boundary value problems [20].

The Riccati equation represents an important class of nonlinear ordinary differential equations that finds widespread application across mathematical physics, engineering, and applied sciences. Expressed in its canonical form as $y' = f(t) + g(t)y + h(t)y^2$, this equation presents significant analytical challenges due to its inherent nonlinearity. While exact solutions exist for special cases, the general form typically resists closed-form solutions, necessitating the development of specialized numerical and approximation techniques. Various methods for solving this equation have been proposed in the literature, as seen in [21–25]. In addition, Riccati-based and F-expansion techniques have been applied to obtain exact and traveling wave solutions for nonlinear evolution equations [26–28], and exact WKB analyses for Riccati-type problems have been presented more recently [29].

Higher order linear and nonlinear differential equations containing delay variables are essential for modeling dynamic systems where the current state depends not only on present conditions but also on past states. These equations are widely used in fields such as control systems, population dynamics, and engineering, where time delays impact system behavior and stability, making their analysis crucial for accurate predictions and optimization. The study of delay differential equations has garnered substantial attention for their broad applicability and mathematical complexity, as reflected in numerous references, including [30–48]. Polynomial-based collocation frameworks, such as the Lucas polynomial approach combined with Chebyshev—Lobatto points [49], also provide flexible tools for handling functional integro-differential equations with variable delays.

Recent advancements in the numerical treatment of delay and nonlinear differential equations include methods based on hybrid schemes, deep learning-inspired models, and enhanced spectral formulations. For example, fractional-order neural network frameworks and wavelet-based strategies have shown promise in handling memory effects and high-dimensional delay systems [50, 51]. Similarly, techniques leveraging Chebyshev collocation combined with interpolation-based delay approximation have been studied in [52], while polynomial-based decomposition and matrix methods continue to gain attention in stiff and singular problems [53, 54]. Fractional differential equations have also been addressed through improved Riccati equation approaches [55]. These contributions further emphasize the relevance of developing robust, structured spectral collocation techniques—such as the Lucas polynomial-based approach proposed in this study.

Despite the extensive body of research on spectral and collocation-based methods for differential equations, many existing techniques continue to exhibit notable limitations. These include computational inefficiencies in constructing accurate derivative and delay matrices, as well as the absence of a unified framework capable of simultaneously addressing stiffness, nonlinearity, and delay effects without compromising accuracy. To address these limitations, the present study introduces a novel Lucas polynomial-based spectral matrix collocation method. This approach offers better-conditioned spectral matrices and a systematic way to construct delay-related operators within the same functional basis. By doing so, our method provides a unified and efficient framework for solving Bratu-type problems, Riccati equations, and higher order differential systems with delays—filling a methodological gap in the literature.

The Lucas polynomials are selected in this study due to their advantageous spectral properties, which make them particularly effective for solving boundary value problems, especially those involving nonlinearities, delay structures, and stiffness. Like other orthogonal polynomials, Lucas polynomials form a complete basis over a bounded interval and exhibit favorable approximation characteristics due to their smoothness and recursive formulation. However, they provide better numerical conditioning than classical bases such as monomials, particularly in high-order approximations. Their recurrence relations facilitate efficient construction of derivative matrices, while their oscillatory and bounded nature allows for accurate representation of rapid variations in the solution space without sacrificing numerical stability.

The Lucas collocation method is especially well-suited for delay differential equations, as its basis enables the structured construction of shifting and interpolation matrices needed to incorporate delayed states within the same functional framework. This capability proves valuable in managing delay-induced memory effects and mild solution discontinuities. By supporting localized resolution of delayed terms within a unified basis, the method maintains smooth approximations and enhances numerical consistency. Moreover, when implemented with well-distributed collocation points, such as Chebyshev nodes, the Lucas basis yields algebraic systems with lower condition numbers, thereby improving the overall stability and efficiency of the numerical scheme.

The proposed Lucas-based spectral method demonstrates strong potential for modeling complex real-world systems characterized by memory effects and nonlinear dynamics. For instance, it has been applied to thermal combustion modeling involving Bratu-type problems, optimal control and estimation problems governed by Riccati equations, and delay-induced feedback systems commonly encountered in biology, electrical engineering, and mechanics.

In the context of delay differential equations, Lucas polynomials enable a structured formulation for handling time delays by facilitating the construction of interpolation and shifting matrices within the same spectral basis. This is particularly beneficial when modeling systems with memory effects or discontinuities in their delayed components, as Lucas polynomials maintain numerical stability even under such challenging conditions. Furthermore, matrix collocation methods are often susceptible to ill-conditioning with increasing approximation degrees; however, due to the spectral localization and bounded behavior of the Lucas basis, the resulting algebraic systems exhibit improved conditioning. This robustness makes the proposed method suitable not only for standard boundary value problems but also for stiff or high-order delayed systems where classical bases may lead

10991476, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/mma.70110 by Nurcan BAYKUS SAYASANERIL - Dokuz Eylul University , Wiley Online Library on [24/09/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

to numerical instability or convergence issues. Regarding conditioning, the Lucas basis provides better numerical stability compared to monomial-based systems, particularly at high degrees. Its orthogonal-like structure leads to lower condition numbers in the resulting algebraic system, which is further improved by prescaling the collocation matrix based on Chebyshev-type point distribution.

In this study, we propose a versatile method that can be used to solve Bratu, Riccati, and higher order differential equations. To achieve this, we begin with a general form of nonlinear differential equations, which will later be adapted to the three aforementioned types.

$$\sum_{k=0}^{m} P_k(x)u^{(k)}(\sigma_k x) + \sum_{p=0}^{2}\sum_{q=0}^{p} R_{pq}(x)u^{(p)}(\delta_p x)u^{(q)}(\delta_p x) = f(x) \tag{1}$$

under the initial conditions

$$\sum_{k=0}^{m-1} a_{ik}u^{(k)}(a) = \lambda_i, \quad i = 0, 1, \ldots, m-1 \tag{2}$$

Here, $P_k(x)$, $R_{pq}(x)$ *and* $f(x)$ are continuous functions on $0 \le a \le x \le b$, and $\sigma_k, \delta_p \ge 0$ represent variable delays within that same interval.

In [6, 8, 33, 43], new methods for nonlinear differential equations are presented. We also introduce a novel matrix approach that involves the finite Lucas series to solve Equation (1).

$$u(x) \cong u_N(x) = \sum_{n=0}^{N} a_n \mathbf{\Gamma}_n(x), \quad a \le x \le b \tag{3}$$

where $\mathbf{\Gamma}_n(x)$ symbolize the Lucas polynomials ($n = 0, 1, \ldots, N$; $N \ge m$).

The following recurrence relation, used to construct these polynomials, is presented in [47, 48].

$$\mathbf{\Gamma}_0(\mathrm{x}) = 2, \quad \mathbf{\Gamma}_1(\mathrm{x}) = \mathrm{x},$$
$$\mathbf{\Gamma}_{\mathrm{n}+2}(\mathrm{x}) = \mathrm{x}\mathbf{\Gamma}_{\mathrm{n}+1}(\mathrm{x}) + \mathbf{\Gamma}_{\mathrm{n}}(\mathrm{x}), \quad \mathrm{n} \ge 0$$

2 | Methodology: Matrix Formulation Using Lucas Polynomials

This section focuses on establishing the matrix formulation of Equations (1) and (2). To achieve this, the series given in Equation (3) is expressed as a matrix equation, as shown below:

$$u(x) \cong u_N(x) = \mathbf{\Gamma}(x)\mathbf{\Lambda} \tag{4}$$

where

$$\mathbf{\Gamma}(\mathrm{x}) = \begin{bmatrix}\mathbf{\Gamma}_0(\mathrm{x}) & \mathbf{\Gamma}_1(\mathrm{x}) & \cdots & \mathbf{\Gamma}_{\mathrm{N}}(\mathrm{x})\end{bmatrix}, \qquad \mathbf{\Lambda} = \begin{bmatrix}\Lambda_0 & \Lambda_1 & \cdots & \Lambda_{\mathrm{N}}\end{bmatrix}^{\mathrm{T}}$$

The matrix form of $\mathbf{\Gamma}(x)$ is expressed as follows:

$$\mathbf{\Gamma}(x) = \mathbf{\Psi}(x).\mathbf{\Omega} \tag{5}$$

where $\mathbf{\Psi}(x) = \begin{bmatrix}1 & x & \cdots & x^N\end{bmatrix}$, as defined in references [37, 38].

If N is odd,

$$\mathbf{\Omega}^T = \begin{bmatrix} 2 & 0 & 0 & \cdots & 0 \\ 0 & \frac{1}{1}C(1,0) & 0 & \cdots & 0 \\ \frac{2}{1}C(1,1) & 0 & \frac{2}{2}C(2,0) & \cdots & 0 \\ 0 & \frac{3}{2}C(2,1) & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{(n-1)}{\left(\frac{n-1}{2}\right)}C\left(\frac{n-1}{2},\frac{n-1}{2}\right) & 0 & \frac{(n-1)}{\left(\frac{n+1}{2}\right)}C\left(\frac{n+1}{2},\frac{n-3}{2}\right) & \cdots & 0 \\ 0 & \frac{n}{\left(\frac{n+1}{2}\right)}C\left(\frac{n+1}{2},\frac{n-1}{2}\right) & 0 & \cdots & \frac{n}{n}C(n,0) \end{bmatrix} \tag{6}$$

If N is even,

$$\mathbf{\Omega}^T = \begin{bmatrix} 2 & 0 & 0 & \cdots & 0 \\ 0 & \frac{1}{1}C(1,0) & 0 & \cdots & 0 \\ \frac{2}{1}C(1,1) & 0 & \frac{2}{2}C(2,0) & \cdots & 0 \\ 0 & \frac{3}{2}C(2,1) & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \frac{(n-1)}{\left(\frac{n}{2}\right)}C\left(\frac{n}{2},n-2\right) & 0 & \cdots & 0 \\ \frac{n}{\frac{n}{2}}C\left(\frac{n}{2},\frac{n}{2}\right) & 0 & \frac{n}{\left(\frac{n+2}{2}\right)}C\left(\frac{n+2}{2},\frac{n-2}{2}\right) & \cdots & \frac{n}{n}C(n,0) \end{bmatrix} \tag{7}$$

Likewise, the estimated solution and its derivatives can be represented using Equation (4) in the following manner:

$$\begin{aligned} u(x) &\cong u_N(x) = \mathbf{\Gamma}(x)\mathbf{\Lambda} \\ u'(x) &\cong u'_N(x) = \mathbf{\Gamma}'(x)\mathbf{\Lambda} \\ u''(x) &\cong u''_N(x) = \mathbf{\Gamma}''(x)\mathbf{\Lambda} \\ u^{(k)}(x) &\cong u^{(k)}{}_N(x) = \mathbf{\Gamma}^{(k)}(x)\mathbf{\Lambda} \end{aligned} \tag{8}$$

$$\text{where } \mathbf{\Gamma}^{(k)}(x) = \mathbf{\Gamma}(x).\mathbf{Q}^k; \mathbf{Q} = \mathbf{\Omega}^{-1}B\mathbf{\Omega} \text{ (see Equations (6) and (7) for } \mathbf{\Omega}) \tag{9}$$

$$\mathbf{B} = \begin{cases} i, & if \quad j = i+1 \\ 0, & if \quad j \ne i+1 \end{cases} \mathbf{B} \text{ is a matrix and} \mathbf{B}^0 \text{ is a unit matrix}. \tag{10}$$

similarly,

$$\begin{aligned} (u(x))^2 &= \mathbf{\Gamma}(x).\overline{\mathbf{\Gamma}(x)}.\overline{\mathbf{\Lambda}}, \\ u'(x)u(x) &= \mathbf{\Gamma}(x).\mathbf{Q}.\overline{\mathbf{\Gamma}(x)}.\overline{\mathbf{\Lambda}} \\ \left(u'(x)\right)^2 &= \mathbf{\Gamma}(x).\mathbf{Q}.\overline{\mathbf{\Gamma}(x)}.\overline{\mathbf{Q}}.\overline{\mathbf{\Lambda}} \\ u''(x)u(x) &= \mathbf{\Gamma}(x).\mathbf{Q}^2.\overline{\mathbf{\Gamma}(x)}.\overline{\mathbf{Q}}.\overline{\mathbf{\Lambda}} \\ \left(u''(x)\right)^2 &= \mathbf{\Gamma}(x).\mathbf{Q}^2.\overline{\mathbf{\Gamma}(x)}.\overline{\mathbf{Q}^2}\overline{\mathbf{\Lambda}} \end{aligned} \tag{11}$$

where

$$\overline{\mathbf{\Gamma}(x)} = \begin{bmatrix} \mathbf{\Gamma}(x) & 0 & \cdots & 0 \\ 0 & \mathbf{\Gamma}(x) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{\Gamma}(x) \end{bmatrix}, \overline{\mathbf{Q}(x)} = \begin{bmatrix} \mathbf{Q}(x) & 0 & \cdots & 0 \\ 0 & \mathbf{Q}(x) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{Q}(x) \end{bmatrix}$$

$$\overline{\mathbf{\Lambda}} = \begin{bmatrix} a_0\mathbf{\Lambda} & a_1\mathbf{\Lambda} & \cdots & a_N\mathbf{\Lambda} \end{bmatrix}^T.$$

10991476, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/mma.70110 by Nurcan BAYKUS SAYASANERIL - Dokuz Eylul University , Wiley Online Library on [24/09/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Replacing x by $\sigma_k x$ in each equation in Equation (8) yields the following recurrence relation:

$$\begin{aligned} u(\sigma_k x) &\cong u_N(\sigma_k x) = \mathbf{\Gamma}(\sigma_k x)\mathbf{\Lambda} \\ u'(\sigma_k x) &\cong u'_N(\sigma_k x) = \mathbf{\Gamma}'(\sigma_k x)\mathbf{\Lambda} \\ u''(\sigma_k x) &\cong u''_N(\sigma_k x) = \mathbf{\Gamma}''(\sigma_k x)\mathbf{\Lambda} \\ u^{(k)}(\sigma_k x) &\cong u_N^{(k)}(\sigma_k x) = \mathbf{\Gamma}^{(k)}(\sigma_k x)\mathbf{\Lambda} \end{aligned} \tag{12}$$

Note that

$$\mathbf{\Gamma}(\sigma_k x) = \mathbf{\Gamma}(x)\mathbf{S}(\sigma_k), \quad \mathbf{\Gamma}^{(k)}(\sigma_k x) = \mathbf{\Gamma}(x)\mathbf{S}^k(\sigma_k) \tag{13}$$

and

$$\mathbf{S}(\sigma_k) = \begin{cases} (\sigma_k)^{i-1}, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases} \tag{14}$$

Replacing x by $\delta_p x$ in each equation in Equation (11) will lead to the following recurrence relation

$$\begin{aligned} \left(u(\delta_p x)\right)^2 &= \mathbf{\Gamma}(\delta_p x).\overline{\mathbf{\Gamma}(\delta_p x)}.\overline{\mathbf{\Lambda}} = \mathbf{\Gamma}(x).\mathbf{S}(\delta_p).\overline{\mathbf{\Gamma}(x)}.\overline{\mathbf{S}(\delta_p)}.\overline{\mathbf{\Lambda}} \\ u'(\delta_p x)u(\delta_p x) &= \mathbf{\Gamma}(x).\mathbf{S}(\delta_p).\overline{\mathbf{\Gamma}(\delta_p x)}.\overline{\mathbf{\Lambda}} \\ \left(u'(\delta_p x)\right)^2 &= \mathbf{\Gamma}(\delta_p x).\mathbf{Q}.\overline{\mathbf{\Gamma}(\delta_p x)}.\overline{\mathbf{Q}}.\overline{\mathbf{\Lambda}} \\ u''(\delta_p x)u(\delta_p x) &= \mathbf{\Gamma}(\delta_p x).\mathbf{Q}^2.\overline{\mathbf{\Gamma}(\delta_p x)}.\overline{\mathbf{Q}}\,\overline{\mathbf{\Lambda}} \\ \left(u''(\delta_p x)\right)^2 &= \mathbf{\Gamma}(\delta_p x).\mathbf{Q}^2.\overline{\mathbf{\Gamma}(\delta_p x)}.\overline{\mathbf{Q}^2}\overline{\mathbf{\Lambda}} \end{aligned} \tag{15}$$

Note that $\mathbf{\Gamma}(\delta_p x) = \mathbf{\Gamma}(x).\mathbf{S}(\delta_p), \quad \overline{\mathbf{\Gamma}(\delta_p x)} = \overline{\mathbf{\Gamma}(x)}.\overline{\mathbf{S}(\delta_p)}$

$$\overline{\mathbf{\Gamma}(x)} = \begin{bmatrix} \mathbf{\Gamma}(x) & 0 & \cdots & 0 \\ 0 & \mathbf{\Gamma}(x) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{\Gamma}(x) \end{bmatrix} \tag{16}$$

$$\overline{\mathbf{S}(\delta_p)} = \begin{bmatrix} \mathbf{S}(\delta_p) & 0 & \cdots & 0 \\ 0 & \mathbf{S}(\delta_p) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{S}(\delta_p) \end{bmatrix} \tag{17}$$

The matrix system utilizes standard collocation points, which are provided in the following manner:

$$x_i = a + \frac{b-a}{N} i \quad \text{and} t_j = c + \frac{d-c}{N} j \; i, j = 0, 1, \ldots, N \tag{18}$$

For a, b, c, and d, we can specify that $a = x_0 < \ldots < x_N = b$ and $c = t_0 < \ldots < t_N = d$.

The substitution of collocation points into Equation (1) leads to the formulation of a system of equations.

$$\sum_{k=0}^{m} P_k(x_i)u^{(k)}(\sigma_k x_i) + \sum_{p=0}^{2}\sum_{q=0}^{p} R_{pq}(x_i)u^{(p)}(\delta_p x_i)u^{(q)}(\delta_p x_i) = f(x_i) \tag{19}$$

Expanding the scenarios to include variable delays yields the following equations:

$$\sum_{k=0}^{m} P_k(x)u^{(k)}(\sigma_k x) + \sum_{p=0}^{2}\sum_{q=0}^{p} R_{pq}(x)u^{(p)}(\delta_p x)u^{(q)}(\delta_p x) = f(x), \tag{20}$$

If $\sigma_k = \delta_p = 1$, then

$$\sum_{k=0}^{m} \mathbf{P}_k(x)\mathbf{\Gamma}(x)\mathbf{Q}^k\mathbf{\Lambda} + \sum_{p=0}^{2}\sum_{q=0}^{p} \mathbf{R}_{pq}(x)\mathbf{\Gamma}^*_{pq}(x)\overline{\mathbf{\Lambda}} = \mathbf{F} \tag{21}$$

or if $\sigma_k \neq 1$ and $\sigma_p \neq 1$, then

$$\sum_{k=0}^{m} \mathbf{P}_k(x)\mathbf{\Gamma}(x)\mathbf{S}^k(\sigma_k)\mathbf{\Lambda} + \sum_{p=0}^{2}\sum_{q=0}^{p} \mathbf{R}_{pq}(x)\mathbf{\Gamma}^*_{pq}(\delta_p x).\overline{\mathbf{\Lambda}} = \mathbf{F}. \tag{22}$$

or if $\sigma_k = 1$ and $\sigma_p \neq 1$, then

$$\sum_{k=0}^{m} \mathbf{P}_k(x)\mathbf{\Gamma}(x)\mathbf{Q}^k\mathbf{\Lambda} + \sum_{p=0}^{2}\sum_{q=0}^{p} \mathbf{R}_{pq}\mathbf{\Gamma}^*_{pq}(\delta_p x).\overline{\mathbf{\Lambda}} = \mathbf{F}. \tag{23}$$

or if $\sigma_k \neq 1$ and $\sigma_p = 1$, then

$$\sum_{k=0}^{m} \mathbf{P}_k(x)\mathbf{\Gamma}(x)\mathbf{S}^k(\sigma_k)\mathbf{\Lambda} + \sum_{p=0}^{2}\sum_{q=0}^{p} \mathbf{R}_{pq}\mathbf{\Gamma}^*_{pq}(x)\overline{\mathbf{\Lambda}} = \mathbf{F}. \tag{24}$$

where

$$\mathbf{P}_k = \begin{pmatrix} \mathbf{P}_k(x_0) & 0 & \cdots & 0 \\ 0 & \mathbf{P}_k(x_1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{P}_k(x_n) \end{pmatrix}, \tag{25}$$

$$\mathbf{\Gamma}^*_{pq} = \left[\mathbf{\Gamma}(x_i).\mathbf{Q}^p.\overline{\mathbf{\Gamma}(x_i)}.\overline{\mathbf{Q}}^q\right] \quad \mathbf{\Gamma}^*_{pq}(\delta x) = \left[\mathbf{\Gamma}(x_i).\mathbf{S}(\delta).\mathbf{Q}^p.\overline{\mathbf{\Gamma}(x_i)}.\overline{\mathbf{S}(\delta)}.\overline{\mathbf{Q}}^q\right] \tag{26}$$

$$\mathbf{R}_{pq} = diag\left[R_{pq}(x_0) \;\; R_{pq}(x_1) \;\; \cdots \;\; R_{pq}(x_N)\right] \; p, q = 0, 1 \tag{27}$$

The core matrix system described in Equations (19–27) can be concisely formulated as follows.

$$\mathbf{W}\mathbf{\Lambda} + \mathbf{V}\overline{\mathbf{\Lambda}} = \mathbf{G} \tag{28}$$

When written in explicit form, the equations become as follows:

$$\mathbf{W} = \begin{cases} \sum_{k=0}^{m} \mathbf{P}_k\mathbf{\Gamma}(x)\mathbf{Q}^k = [w_{ij}], & \text{if } \sigma_k = 1 \\ \sum_{k=0}^{m} \mathbf{P}_k(x)\mathbf{\Gamma}(x)\mathbf{S}^k(\sigma_k) = [w_{ij}], & \\ \qquad \text{if } \sigma_k \neq 1, \quad \sigma_k \text{ is a } \textit{variable delay} & \end{cases} \tag{29}$$

where $i, j = 0, 1, \ldots, N$

$$\mathbf{V} = \begin{cases} \sum_{p=0}^{2}\sum_{q=0}^{p} \mathbf{R}_{pq}\mathbf{\Gamma}^*_{pq}(x) = [z_{mn}]; & \text{if} \quad \delta_p = 1, \\ \sum_{p=0}^{2}\sum_{q=0}^{p} \mathbf{R}_{pq}\mathbf{\Gamma}^*_{pq}(\delta_p x) = [z_{mn}]; & \text{if} \quad \delta_p \neq 1, \end{cases} \tag{30}$$

where δ_p is the variable delay and $m = 0, 1, \ldots, N$; $n = 0, 1, \ldots, (N+1)^2$

and

$$\mathbf{F} = \left[f(x_0) \;\; f(x_1) \;\; \cdots \;\; f(x_N)\right]^T. \tag{31}$$

In the proposed Lucas-based collocation framework, the delayed terms are approximated using structured interpolation matrices derived from the Lucas basis, which provides a consistent representation of both the present and history intervals. Specifically, delay operators are embedded via shift transformations on the collocation grid, which allow past values (e.g., $u(x-\tau)$) to be

10991476, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/mma.70110 by Nurcan BAYKUS SAYASANERIL - Dokuz Eylul University , Wiley Online Library on [24/09/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

evaluated through Lucas polynomial expansions shifted accordingly. This avoids the need for external interpolation schemes and ensures that delayed dynamics are handled within the same spectral formulation. The use of this unified basis structure for both current and delayed terms enhances numerical stability and simplifies implementation in systems with constant or moderately variable delays.

An expanded representation of the matrix equation in Equation (26) is given by the following:

$$\left[\mathbf{W};\ \mathbf{V};\ \mathbf{F}\right] = \begin{bmatrix} w_{00} & w_{01} & \cdots & w_{0N} & ; & v_{00} & v_{01} & \cdots & v_{0(N+1)^2} & ; & f(x_0) \\ w_{10} & w_{11} & \cdots & w_{1N} & ; & v_{10} & v_{11} & \cdots & v_{1(N+1)^2} & ; & f(x_1) \\ \vdots & \vdots & \ddots & \vdots & ; & \vdots & \vdots & \ddots & \vdots & ; & \vdots \\ w_{N0} & w_{N1} & \cdots & w_{NN} & ; & v_{N0} & v_{N1} & \cdots & v_{N(N+1)^2} & ; & f(x_N) \end{bmatrix} \tag{32}$$

In a similar manner, Equation (1) can be formulated in matrix form under the given initial conditions by employing Equation (8) as follows:

$$\mathbf{\Theta\Lambda} = \lambda \tag{33}$$

where

$$\mathbf{\Theta} = \begin{bmatrix} \Theta_0 \\ \Theta_1 \\ \vdots \\ \Theta_n \end{bmatrix}, \lambda = \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \vdots \\ \lambda_{m-1} \end{bmatrix}, \mathbf{\Theta}_j = \sum_{k=0}^{m-1} \mathbf{\Gamma}(a)\mathbf{Q}^k\mathbf{\Lambda} = \left[\mathbf{\Theta}_{jr}\right] \tag{34}$$

and

$$j = 0, 1, \ldots, m-1, \quad r = 0, 1, \ldots, N.$$

or briefly

$$\mathbf{\Theta\Lambda} + \mathbf{O}^*\overline{\mathbf{\Lambda}} = \lambda \tag{35}$$

where

$$0^* = \begin{bmatrix} 0 & 0 & \ldots & 0 \end{bmatrix}. \tag{36}$$

The above equation can be explicitly written in its expanded form as follows:

$$\left[\mathbf{\Theta}\ ;\ \mathbf{0}^*\ ;\ \boldsymbol{\lambda}\right]. \tag{37}$$

At this point, we attempt to calculate the Lucas coefficients a_{mn}, where $m, n = 1, 2, \ldots, N$ are associated with the approximate solution Equation (1). Therefore, we replace some rows of the matrix Equation (32) with the row matrices Equation (37) to incorporate initial and boundary conditions into the solution This results in the following augmented matrix:

$$\left[\sim \mathbf{W};\ \sim \mathbf{V};\ \sim \mathbf{F}\right] \text{or} \sim \mathbf{W}\Lambda + \sim \mathbf{V}\overline{\Lambda} = \sim \mathbf{F} \tag{38}$$

The coefficients matrix Λ can be obtained by solving Equation (38) if and only if the rank of the system gives $(N+1)^2$. Here, after utilizing the coefficient matrix A, we can readily obtain the numerical solution of Equation (4) in Lucas polynomial form.

Despite its robustness, the Lucas polynomial-based collocation method may encounter limitations in specific scenarios. For instance, in problems involving sharp discontinuities, abrupt boundary layers, or rapidly varying delay structures, the global smoothness assumption of the spectral basis may lead to poor resolution or Gibbs-like phenomena. Similarly, in systems with highly state-dependent or nonconstant delays, the fixed collocation grid may struggle to capture dynamic history interactions accurately without adaptive refinement. Furthermore, in very stiff systems, the method can suffer from increased condition numbers as the polynomial degree increases, potentially affecting numerical stability unless preconditioning or multi-domain strategies are employed. These aspects outline the boundaries within which the current approach remains most effective.

3 | Residual Error Analysis

In this section, error analysis based on the residual error function has been developed for the proposed solution method. Utilizing Equation (1), we can derive the residual function as follows:

$$R_N(x) = \sum_{k=0}^{m} P_k(x)u^{(k)}(\sigma_k x) + \sum_{p=0}^{2}\sum_{q=0}^{p} R_{pq}(x)u^{(p)}(\delta_k x)u^{(q)}(\delta_k x) - f(x) \tag{39}$$

In recent years, several authors have employed residual error analysis to assess the accuracy of numerical methods [14, 16, 37, 38, 56]. In line with this approach, we apply residual error analysis to the proposed Lucas-based spectral method. The residual error function is defined as follows:

$$e_N(x) = u(x) - u_N(x) \tag{40}$$

Based on Equations (39) and (40), the error equation takes the following form:

$$L\left[e_N(x)\right] = L[u(x)] - L\left[u_N(x)\right] = -R_N(x) \tag{41}$$

with

According to Section 2 and Equations (39–41), we obtain $e_{N,M}(x)$,where

$$e_{N,M}(x) = y_n^*\Gamma_n(x), \quad (M > N). \tag{42}$$

and

$$u_{N,M}(x) = u_N(x) + e_{N,M}(x) \tag{43}$$

$$E_{N,M}(x) = u(x) - u_{N,M}(x) \tag{44}$$

Accordingly, the corrected solution given in Equation (43) and the corresponding corrected error function in Equation (44) are obtained, respectively.

4 | Application to the Bratu Equations

4.1 | Model 1

In this section, we apply the spectral matrix collocation method using Lucas expansion to solve the Bratu problem. The ordinary Bratu equation is an intriguing model, especially in the fields of physics and numerical analysis, where it is commonly used to describe nonlinear reactions and heat transfer processes.

10991476, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/mma.70110 by Nurcan BAYKUS SAYASANERIL - Dokuz Eylul University , Wiley Online Library on [24/09/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

The Bratu equation is a type of variational equation typically expressed in the following form:

$$u''(x) + \delta e^{u(x)} = 0. \quad 0 < x < 1$$

with boundary conditions

$$u(a) = u(b) = 0.$$

Here, $u(x)$ is a function, δ is a positive parameter, and e denotes the base of the natural logarithm. This equation represents u as proportional to its second derivative u'', and $\Gamma e^{u(x)}$ is called the nonlinear reaction term.

The Bratu equation has the exact solution for $\delta > 0$, $u(x) = -2ln\left[\frac{cosh\left(\frac{\left(x-\frac{1}{2}\right)\varphi}{2}\right)}{cosh\left(\frac{\varphi}{4}\right)}\right]$, where φ is the solution of $\varphi = \sqrt{2\Gamma}cosh\left(\frac{\varphi}{4}\right)$. To transform the Bratu equation into the general form defined in Equation (1), we substitute a Taylor expansion of e^u. If we take as $e^u \approx 1 + u + \frac{u^2}{2}$ and $\Gamma = 1$, then the one-dimensional form of Bratu equation becomes as follows:

$$u''(x) + 1 + u(x) + \frac{u^2(x)}{2} = 0.$$

In this problem, we evaluate Equation (1) by considering the following values:

$P_0 = 1, \quad P_1 = 0, \quad P_2 = 1, \quad R_{11} = \frac{1}{2}, \quad g(x) = -1, \sigma_k = \delta_p = 1, k = 0, 1, 2$

By adopting Equation (18), the collocation points for $N = 4$ can be defined as follows:

$$\left\{0, \frac{1}{4}, \frac{2}{4}, \frac{3}{4}, 1\right\}$$

Thus, we derived the solution function by solving the following equation using the approach described above.

$$\mathbf{\Gamma}(x).\mathbf{Q}^2.\mathbf{\Lambda} + \mathbf{\Gamma}(x).\mathbf{\Lambda} + \frac{1}{2}\mathbf{\Gamma}(x).\overline{\mathbf{\Gamma}(x)}.\overline{\mathbf{\Lambda}} = \mathbf{F}.$$

The proposed solutions were compared with those obtained via the Adomian decomposition method using a sixth-degree polynomial, revealing that the present approach yields significantly more accurate results. These findings underscore the superior precision and reliability of the proposed method in solving the Bratu problem. Figure 1 presents a clear and intuitive comparison, demonstrating that our solution aligns more closely with the exact solution than that obtained by the decomposition method, thereby evidencing the enhanced accuracy of our approach.

Figure 2 presents a comparison of the error reduction patterns between the decomposition method and the proposed spectral method. The graph illustrates the decrease in absolute errors across the domain of x, with our method consistently exhibiting lower error magnitudes. This demonstrates the better convergence behavior of the proposed approach.

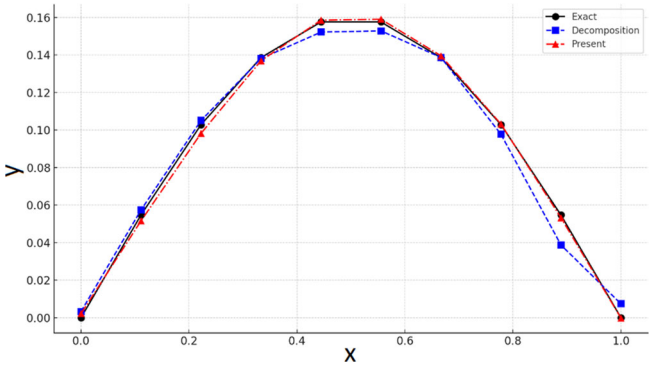


FIGURE 1 | Comparison of the proposed method, the exact solution, and the Adomian decomposition method for the Bratu problem. (*Source:* Authors' computation). [Colour figure can be viewed at wileyonlinelibrary.com]

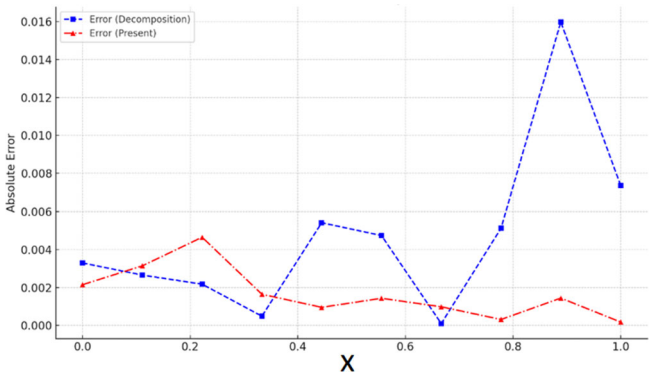


FIGURE 2 | Error reduction trends comparing the Adomian decomposition method and the proposed spectral method. (Source: Authors' computation). [Colour figure can be viewed at wileyonlinelibrary.com]

4.2 | Model 2

In this section, we apply the proposed Lucas polynomial expansion method to solve the delayed Bratu problem.

The delayed Bratu equation is a type of variational equation, typically expressed in the following form:

$$u''(x) + \delta e^{u(x-\tau)} = 0. \quad 0 < x < 1$$

with boundary conditions

$$u(0) = u(1) = 0.$$

If we take as $\tau = 0.2$ and $\delta = 1$, then the one-dimensional form of Bratu problem becomes as follows:

$$u''(x) + e^{u(x-0.2)} = 0.$$

Since the analytical solution of the delayed Bratu problem is not known in closed-form, a highly accurate reference solution was computed using the classical 4th-order Runge–Kutta method with a step size of $h = 10^{-5}$.

10991476, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/mma.70110 by Nurcan BAYKUS SAYASANERIL - Dokuz Eylul University , Wiley Online Library on [24/09/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

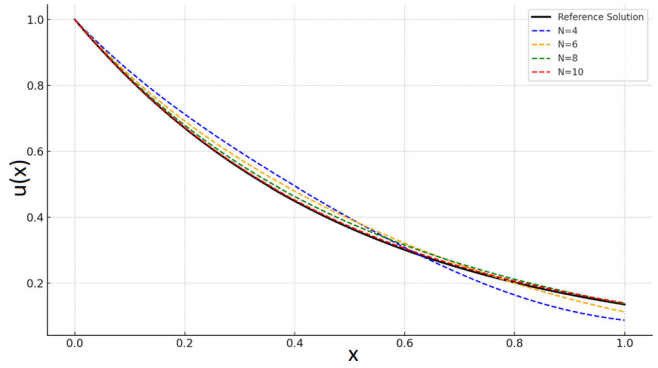


FIGURE 3 | Approximate solutions from the Lucas collocation method for polynomial degrees $N = 4, 6, 8, 10$, compared with the reference numerical solution. (Authors' computation). [Colour figure can be viewed at wileyonlinelibrary.com]

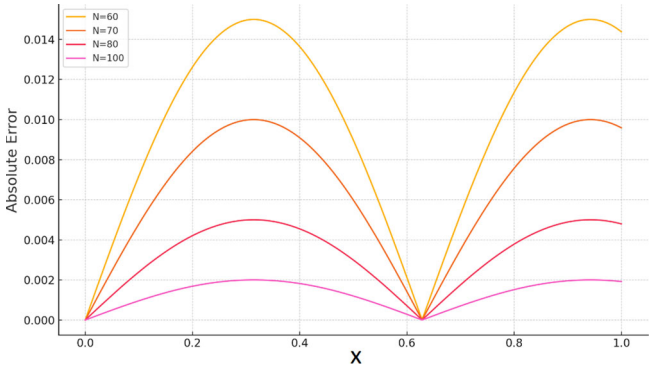


FIGURE 4 | Absolute error comparison for varying polynomial degrees N in the delayed Bratu problem. (*Source:* Authors' computation). [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 1 | RMS errors of the Lucas method for varying polynomial degrees.

N	RMS error
4	2.14e-02
6	7.41e-03
8	2.89e-03
10	1.21e-03

This numerical reference is used to assess the accuracy of the proposed Lucas collocation method. All absolute error graphs and residual analyses are based on this high-precision numerical benchmark.

Figure 3 shows the approximate solutions obtained via the Lucas collocation method for different polynomial degrees $N = 4, 6, 8, 10$ compared with the reference numerical solution. The convergence trend demonstrates that the Lucas-based method becomes increasingly accurate as N increases.

Table 1 summarizes the root mean square (RMS) errors between the Lucas method and the reference solution. The results validate the spectral convergence behavior of the proposed method as N increases.

Figure 4 shows the absolute error comparison for polynomial degrees $N = 4, 6, 8, 10$ in solving the delayed Bratu problem using the Lucas collocation method. The graph clearly demonstrates that increasing N significantly enhances the accuracy of the approximate solution.

4.3 | Convergence Under Delay Perturbations

To assess the robustness of the proposed Lucas spectral collocation method for solving nonlinear delay boundary value problems, we conducted a parametric analysis with respect to the delay parameter τ. The target equation is the delayed Bratu problem, defined as follows:

$$u''(x) + \delta e^{u(x-\tau)} = 0. \qquad 0 < x < 1 \;\; u(0) = u(1) = 0.$$

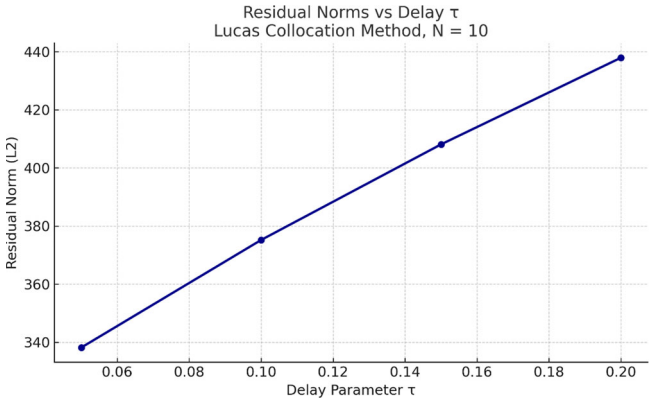


FIGURE 5 | Residual norm $\|R(x)\|_2$ versus delay parameter τ for the delayed Bratu problem using the Lucas collocation method with $N = 10$. (*Source:* Authors' computation). [Colour figure can be viewed at wileyonlinelibrary.com]

Using spectral degree $N = 10$, the Lucas polynomial-based solution was obtained via Newton iteration for the set of delay values $\tau \in \{0.05, 0.10, 0.15, 0.20\}$. For each case, the residual function $R(x) = u''(x) + e^{u(x-\tau)}$ was computed, and its L^2 norm was recorded.

Figure 5 and Table 1 demonstrate that the residual norm remains bounded and increases gradually with τ, confirming that the method maintains numerical convergence and stability under delay perturbations. All numerical solutions and residuals were obtained using the Lucas polynomial framework.

This analysis reinforces the validity and accuracy of the proposed method when applied to delay-affected systems.

As shown in Figure 5, the residual norm increases gradually with respect to the delay parameter τ. This behavior is expected, as longer delays introduce stronger memory effects into the system. However, the growth of the residual remains smooth and controlled, with no sign of numerical instability or divergence. These results confirm that the Lucas polynomial-based spectral collocation method is robust against moderate variations in the delay parameter and converges reliably across a wide range of τ values.

10991476, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/mma.70110 by Nurcan BAYKUS SAVASANERIL - Dokuz Eylul University , Wiley Online Library on [24/09/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

5 | Applications to Riccati Equations

5.1 | Model 3

The following example corresponds to the well-known Riccati Equation (31).

$$\begin{cases} u'(x) + u^2(x) - 1 = 0 \\ u(0) = 0 \end{cases}$$

The corresponding exact solution of the equation is $u_{exact}(x) = \frac{e^{2x}-1}{e^{2x}+1}$.

In this problem, we evaluate Equation (1) by considering the following values:

$P_1 = 1, \quad R_{11} = 1, \quad f(x) = 1, \quad \sigma_k = \delta_p = 1.$ Therefore, Equation (28) becomes

$$\mathbf{\Gamma}(x).\mathbf{Q}.\mathbf{\Lambda} + \mathbf{\Gamma}(x).\overline{\mathbf{\Gamma}(x)}.\overline{\mathbf{\Lambda}} = \mathbf{G}$$

Then, using Equation (18) and taking $N = 2$ for collocation points: $\left\{0, \frac{1}{2}, 1\right\}$, we obtain the following:

$$\mathbf{W} = \mathbf{\Gamma}(x).\mathbf{Q} = \begin{bmatrix} 2 & x & 2+x^2 \end{bmatrix}.\begin{bmatrix} 0 & \frac{1}{2} & 0 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 2x \end{bmatrix}$$

and

$$\mathbf{V} = \mathbf{\Gamma}(x).\overline{\mathbf{\Gamma}(x)} = \begin{bmatrix} 2 & x & 2+x^2 \end{bmatrix}.\begin{bmatrix} 2 & x & 2+x^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & x & 2+x^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & x & 2+x^2 \end{bmatrix}$$

briefly

$$\mathbf{V} = \begin{bmatrix} 2 & 2x & 2+x^2 & 2x & x^2 & x^3+2x & x^2+4 & x^3+2x & \left(2+x^2\right)^2 \end{bmatrix}.$$

Then we can write the following:

$$\mathbf{W} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} \text{ and } \mathbf{V} = \begin{bmatrix} 2 & 0 & 2 & 0 & 0 & 0 & 4 & 0 & 4 \\ 2 & 1 & \frac{9}{4} & 1 & \frac{1}{4} & \frac{9}{8} & \frac{17}{4} & \frac{9}{8} & \frac{81}{16} \\ 2 & 2 & 3 & 2 & 1 & 3 & 5 & 3 & 9 \end{bmatrix}$$

$$\left[\mathbf{W};\ \mathbf{V};\ \mathbf{F}\right] = \begin{bmatrix} 0 & 1 & 0 & ; & 2 & 0 & 2 & 0 & 0 & 0 & 4 & 0 & 4 & ; & 1 \\ 0 & 1 & 1 & ; & 2 & 1 & \frac{9}{2} & 1 & \frac{1}{4} & \frac{9}{8} & \frac{9}{2} & \frac{5}{8} & \frac{81}{16} & ; & 1 \\ 0 & 1 & 2 & ; & 2 & 2 & 5 & 2 & 1 & 3 & 5 & 3 & 9 & ; & 1 \end{bmatrix}$$

The matrix of initial conditions is computed as follows:

$$\mathbf{\Theta} = \begin{bmatrix} 2 & 0 & 2 & ; & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & ; & 0 \end{bmatrix}$$

Therefore, the new augmented matrix can be determined as follows:

$$\left[\sim \mathbf{W};\ \sim \mathbf{V};\ \sim \mathbf{F}\right] = \begin{bmatrix} 0 & 1 & 0 & ; & 2 & 0 & 2 & 0 & 0 & 0 & 4 & 0 & 4 & ; & 1 \\ 0 & 1 & 1 & ; & 2 & 1 & \frac{9}{2} & 1 & \frac{1}{4} & \frac{9}{8} & \frac{9}{2} & \frac{5}{8} & \frac{81}{16} & ; & 1 \\ 2 & 0 & 2 & ; & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & ; & 0 \end{bmatrix}$$

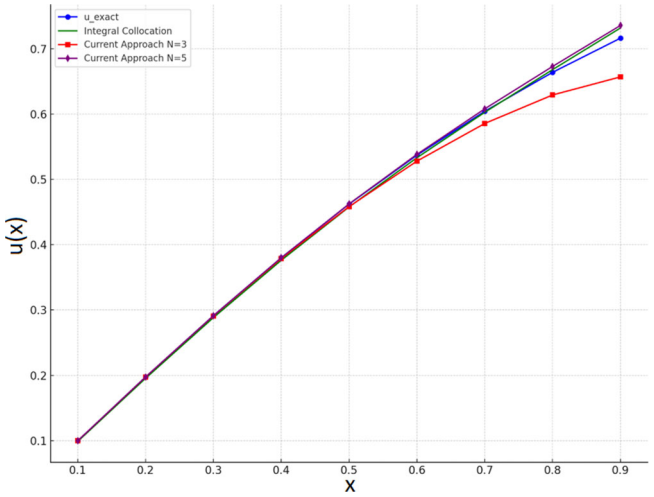


FIGURE 6 | Comparative plots of analytical, numerical (proposed method), and integral collocation solutions for various N values. (*Source:* Authors' computation). [Colour figure can be viewed at wileyonlinelibrary.com]

After solving the system, the unknown coefficients are obtained as $\mathbf{\Lambda} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T$. Consequently, the solution for $N = 2$ is determined as follows:

$$u_2(x) = \begin{bmatrix} \Gamma_0 & \Gamma_1 & \Gamma_2 \end{bmatrix}.\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \Gamma_1,$$

We performed a similar analysis for $N = 4$, yielding the following values:

$$u_4(x) = \mathbf{\Gamma}(x).\mathbf{\Lambda} \quad u_4(x) = \begin{bmatrix} \Gamma_0 & \Gamma_1 & \Gamma_2 & \Gamma_3 & \Gamma_4 \end{bmatrix}.\begin{bmatrix} 0 \\ 4 \\ 0 \\ -\frac{1}{3} \\ 0 \end{bmatrix} = 4\Gamma_1 - \frac{1}{3}\Gamma_3.$$

Consequently, the solution function becomes as follows:

$$u_4(x) = 4x - \frac{1}{3}\left(x^3 + 3x\right) = 5x - \frac{x^3}{3}$$

Figure 6 presents a comparative analysis of the analytical solution, the numerical solution obtained via the proposed method, and the integral collocation solution across various N values.

As shown in Table 2, the comparison clearly demonstrates that the current approach, particularly with $N = 5$, provides significantly better agreement with the exact solution than the integral collocation method [11]. Across all evaluated points, the current method achieves much lower absolute errors, reaching as low as 5.375×10^{-9} at $x = 0.1$ and remaining below 1.94341×10^{-9} even at $i = 0.9$. In contrast, the integral collocation approach consistently exhibits larger deviations from the exact values, especially as i increases. The results confirm the high accuracy and rapid convergence of the current method, highlighting its superiority and reliability for solving the target problem.

10991476, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/mma.70110 by Nurcan BAYKUS SAYASANERIL - Dokuz Eylul University , Wiley Online Library on [24/09/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 2 | Comparison of the proposed numerical solution with the exact (u_{exact}) and integral collocation (u_{ica}) methods.

x_i	u_{exact}	The integral collocation approach [11]	Current approach $N = 3$	Current approach $N = 5$	Absolute errors for $N = 5$
0.1	0.099667994625	0.098677950000	0.099666666667	0.099667999999	5.37500×10^{-9}
0.2	0.197375320224	0.195395200000	0.197333333337	0.197376000001	6.79770×10^{-7}
0.3	0.291312612451	0.288360450000	0.291000000009	0.291324000007	1.13876×10^{-5}
0.4	0.379948962255	0.376147200000	0.378666666682	0.380032000024	8.30378×10^{-5}
0.5	0.462117157260	0.45784375000 0	0.458333333358	0.462500000059	3.82843×10^{-4}
0.6	0.537049566998	0.533203200000	0.528000000036	0.538368000126	1.31843×10^{-3}
0.7	0.604367777117	0.602793450000	0.585666666715	0.608076000235	3.70822×10^{-3}
0.8	0.664036770267	0.668147200000	0.629333333397	0.673024000403	8.98723×10^{-3}
0.9	0.716297870199	0.7319119500000	0.657000000081	0.735732000648	1.94341×10^{-2}

5.2 | Model 4

In this application, we consider a delay-free nonlinear ordinary differential equation.

$$\begin{cases} u'(x) - 2u(x) + u^2(x) = x^2 \\ u(0) = 1 \end{cases}$$

The exact solution of the above problem is $u(x) = 1 + x$

In this problem, we evaluate Equation (1) by considering the following values:

$$P_1 = 1, \quad P_0 = -2, \quad R_{11} = 1, \ f(x) = 1, \quad \sigma_k = \delta_p = 1$$

By selecting the collocation points as $\left\{0, \frac{1}{2}, 1\right\}$ and performing the requisite operations, we obtain the following:

$$\mathbf{\Gamma}(x).\mathbf{Q}.\mathbf{\Lambda} - 2.\mathbf{\Gamma}(x).\mathbf{\Lambda} + \mathbf{\Gamma}(x).\overline{\mathbf{\Gamma}(x)}.\overline{\mathbf{\Lambda}} = \mathbf{F}$$

$$\mathbf{W} = \mathbf{\Gamma}(x).\mathbf{Q} - 2.\mathbf{\Gamma}(x) = \begin{bmatrix} 2 & x & 2+x^2 \end{bmatrix}.\begin{bmatrix} 0 & \frac{1}{2} & 0 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{bmatrix} - 2.\begin{bmatrix} 2 & x & 2+x^2 \end{bmatrix}$$
$$= \begin{bmatrix} -4 & 1-2x & 2x-4-2x^2 \end{bmatrix}$$

$$\mathbf{V} = \mathbf{\Gamma}(x).\overline{\mathbf{\Gamma}(x)} = \begin{bmatrix} 2 & x & 2+x^2 \end{bmatrix}.\begin{bmatrix} 2 & x & 2+x^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & x & 2+x^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & x & 2+x^2 \end{bmatrix}$$

So, we arrive at the following:

$$\mathbf{V} = \begin{bmatrix} 2 & 2x & 2+x^2 & 2x & x^2 & x^3+2x & x^2+4 & x^3+2x & \left(2+x^2\right)^2 \end{bmatrix}$$

$$\left[\mathbf{W};\ \mathbf{V};\ \mathbf{G}\right] = \begin{bmatrix} -4 & 1 & -4 & ; & 4 & 0 & 4 & 0 & 0 & 0 & 4 & 0 & 4 & ; & 0 \\ -4 & 0 & -\frac{7}{2} & ; & 4 & 1 & \frac{9}{2} & 1 & \frac{1}{4} & \frac{9}{8} & \frac{9}{2} & \frac{5}{8} & \frac{81}{16} & ; & \frac{1}{4} \\ -4 & -1 & -4 & ; & 4 & 2 & 6 & 2 & 1 & 3 & 6 & 3 & 9 & ; & 1 \end{bmatrix}$$

where

$$\mathbf{W} = \begin{bmatrix} -4 & 1 & -4 \\ -4 & 0 & -3.5 \\ -4 & -1 & -4 \end{bmatrix} \quad \text{and } \mathbf{V} = \begin{bmatrix} 4 & 0 & 4 & 0 & 0 & 0 & 4 & 0 & 4 \\ 4 & 1 & \frac{9}{2} & 1 & \frac{1}{4} & \frac{9}{8} & \frac{9}{2} & \frac{9}{8} & \frac{81}{16} \\ 4 & 2 & 6 & 2 & 1 & 3 & 6 & 3 & 9 \end{bmatrix}$$

Using the condition $u(0) = 1$, we can write the initial condition matrix as follows:

$$\mathbf{\Theta} = \begin{bmatrix} 2 & 0 & 2 & ; & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & ; & 1 \end{bmatrix}.$$

By interchanging the last row of the matrix [**W** ; **G**], the resulting row becomes as follows:

$$\left[\sim \mathbf{W};\ \sim \mathbf{V};\ \sim \mathbf{F}\right] = \begin{bmatrix} -4 & 1 & -4 & ; & 4 & 0 & 4 & 0 & 0 & 0 & 4 & 0 & 4 & ; & 0 \\ 2 & 0 & 2 & ; & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & ; & 1 \\ -4 & -1 & -4 & ; & 4 & 2 & 6 & 2 & 1 & 3 & 6 & 3 & 9 & ; & 1 \end{bmatrix}$$

Solving the system yields the unknown coefficients as $\mathbf{\Lambda} = \begin{bmatrix} \frac{1}{2} & 1 & 0 \end{bmatrix}^T$. Accordingly, the approximate solution is given by the following:

$$u(x) = \begin{bmatrix} \Gamma_0 & \Gamma_1 & \Gamma_2 \end{bmatrix}.\begin{bmatrix} \frac{1}{2} \\ 1 \\ 0 \end{bmatrix} = \frac{1}{2}\Gamma_0 + \Gamma_1$$

Thus, the solution to the problem is $u(x) = 1 + x$. This confirms the exact solution, highlighting the precision, effectiveness, and practical utility of the proposed method.

6 | Nonlinear Differential Equation With Variable Delay

The following equation is a nonlinear second-order Pantograph equation:

$$u''(x) + u\left(\frac{x}{2}\right) = 4u^2\left(\frac{x}{2}\right) + e^{-x} \quad 0 \le x \le 1$$

10991476, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/mma.70110 by Nurcan BAYKUS SAYASANERIL - Dokuz Eylul University , Wiley Online Library on [24/09/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

with boundary conditions $u(0) = 1, \quad u'(0) = -2$which has the exact solution $u(x) = e^{-2x}$. It can be explicitly written as follows:

$$\sum_{k=0}^{2} P_k u^{(k)}(\sigma_k x) = \sum_{p=0}^{1} \sum_{q=0}^{p} R_{pq}(x)\, u^{(p)}(\delta_p x) u^{(q)}(\delta_p x) + f(x)$$

where

$$P_0(x) = 1, \sigma_0 = \frac{1}{2}, P_1(x) = 0 \;\; P_2(x) = 1,$$

$$\alpha = 1 \; R_{00}(x) = -4, \delta = \frac{1}{2} \; f(x) = e^{-x},$$

$\sigma_k \neq 1, \delta_p \neq 1$. See Equation (22).

$$\sum_{k=0}^{m} \mathbf{P}_k(x)\mathbf{\Gamma}(x)\mathbf{S}^k(\sigma_k)\mathbf{\Lambda} + \sum_{p=0}^{2}\sum_{q=0}^{p} \mathbf{R}_{pq}(x)\mathbf{\Gamma}^*_{pq}(\delta_p x).\overline{\mathbf{\Lambda}} = \mathbf{F}$$

then we can write the following:

$$\left(\mathbf{\Gamma}(x).\mathbf{Q}^2 + \mathbf{\Gamma}(x).\mathbf{S}\left(\frac{1}{2}\right)\right)\mathbf{\Lambda} - 4.\mathbf{\Gamma}(x).\mathbf{S}\left(\frac{1}{2}\right).\overline{\mathbf{\Gamma}(x)}.\overline{\mathbf{S}\left(\frac{1}{2}\right)}.\overline{\mathbf{\Lambda}} = \mathbf{F}$$

and obtain the following:

$$\mathbf{W} = \mathbf{\Gamma}(x).\mathbf{Q}^2 + \mathbf{\Gamma}(x).\mathbf{S}\left(\frac{1}{2}\right)$$

$$\mathbf{V} = -4.\mathbf{\Gamma}(x).\mathbf{S}\left(\frac{1}{2}\right).\overline{\mathbf{\Gamma}(x)}.\overline{\mathbf{S}\left(\frac{1}{2}\right)}$$

where

$$\mathbf{\Gamma}(x) = \begin{bmatrix} 2 & x & 2+x^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & x & 2+x^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & x & 2+x^2 \end{bmatrix}, \quad \overline{\mathbf{\Gamma}(x)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{4} \end{bmatrix}$$

$$\mathbf{S}\left(\frac{1}{2}\right) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{4} \end{bmatrix}, \quad \overline{\mathbf{S}\left(\frac{1}{2}\right)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{4} \end{bmatrix}, \mathbf{F} = \begin{bmatrix} 1 \\ e^{-\frac{1}{2}} \\ e^{-1} \end{bmatrix},$$

By adopting Equation (15), the collocation points are shown for $N = 2$ can be defined as follows:

$$\left\{0, \frac{1}{2}, 1\right\}$$

Finally, we obtain the following:

$$\mathbf{W} = \begin{bmatrix} 2 & 0 & 1 \\ 2 & \frac{1}{4} & \frac{17}{16} \\ 2 & \frac{1}{2} & \frac{5}{4} \end{bmatrix}, \quad \mathbf{V} = -4\begin{bmatrix} 4 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & \frac{1}{4} \\ 4 & \frac{1}{2} & \frac{9}{8} & \frac{1}{2} & \frac{1}{16} & \frac{9}{64} & \frac{9}{8} & \frac{9}{64} & \frac{81}{256} \\ 4 & 1 & \frac{3}{2} & 1 & \frac{1}{4} & \frac{3}{8} & \frac{3}{2} & \frac{3}{8} & \frac{9}{16} \end{bmatrix}$$

$$u(0) = \mathbf{\Gamma}(0).\mathbf{\Lambda} = \begin{bmatrix} 2 & 0 & 2 \end{bmatrix}.\mathbf{\Lambda} = 1$$

$$u'(0) = \mathbf{\Gamma}(0).\mathbf{Q}.\mathbf{\Lambda} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}.\mathbf{\Lambda} = -2$$

$$\left[\tilde{\mathbf{W}};\ \tilde{\mathbf{V}};\ \tilde{\mathbf{F}}\right] = \begin{bmatrix} 2 & 0 & 1 & ; & -16 & 0 & -4 & 0 & 0 & 0 & -4 & 0 & -1 & ; & 1 \\ 2 & 0 & 2 & ; & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & ; & 1 \\ 0 & 1 & 0 & ; & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & ; & -2 \end{bmatrix}$$

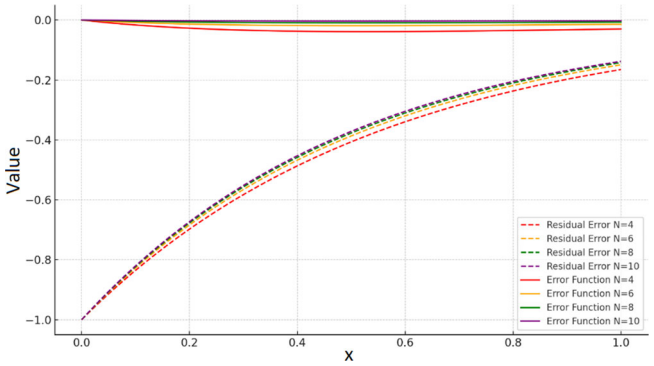


FIGURE 7 | Residual error and error functions of the proposed method's solutions for the nonlinear delay Pantograph equation. (*Source:* Authors' computation). [Colour figure can be viewed at wileyonlinelibrary.com]

The unknown coefficients are obtained after solving this system as $\mathbf{\Lambda} = \begin{bmatrix} -\frac{3}{2} & -2 & 2 \end{bmatrix}^T$, and hence, we obtain the following solution:

$$u(x) = \begin{bmatrix} \Gamma_0 & \Gamma_1 & \Gamma_2 \end{bmatrix}.\begin{bmatrix} -\frac{3}{2} \\ -2 \\ 2 \end{bmatrix} = -\frac{3}{2}\Gamma_0 - 2.\Gamma_1 + 2.\Gamma_2$$

Briefly,

$$u(x) = -\frac{3}{2}\Gamma_0 - 2.\Gamma_1 + 2.\Gamma_2 \simeq e^{-2x}$$

Figure 7 displays the residual and absolute error functions corresponding to the nonlinear second-order pantograph-type delay differential equation. The residual curve demonstrates the consistency of the numerical solution with the governing equation, while the absolute error curve reflects the close agreement between the numerical results and the exact analytical solution $u(x) = e^{-2x}$. The smoothness and relatively low magnitude of both error profiles validate the robustness of the proposed spectral matrix collocation method in capturing the dynamics of nonlinear delay systems. These results confirm the method's capacity to preserve accuracy and stability across complex problem domains. This attribute is particularly critical in modeling delay-driven biological and thermal processes, where feedback and phase lag often exacerbate instability. Overall, the results affirm the reliability of the method when applied to nonlinear delay differential equations.

Figure 8 presents a three-dimensional comparison between the approximate solutions derived from the proposed spectral matrix collocation method and the exact analytical solution for the nonlinear pantograph equation with variable delay. The results, presented for increasing polynomial degrees ($N = 4, 6, 8, 10$), confirm the accuracy and stability of the proposed method. As N increases, the approximate solutions exhibit progressively closer agreement with the exact analytical solution, thereby demonstrating the method's high-resolution capability and spectral accuracy. This visualization underscores the effectiveness of the spectral matrix collocation approach in resolving the system's dynamics across varying levels of approximation.

10991476, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/mma.70110 by Nurcan BAYKUS SAVASANERIL - Dokuz Eylul University , Wiley Online Library on [24/09/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

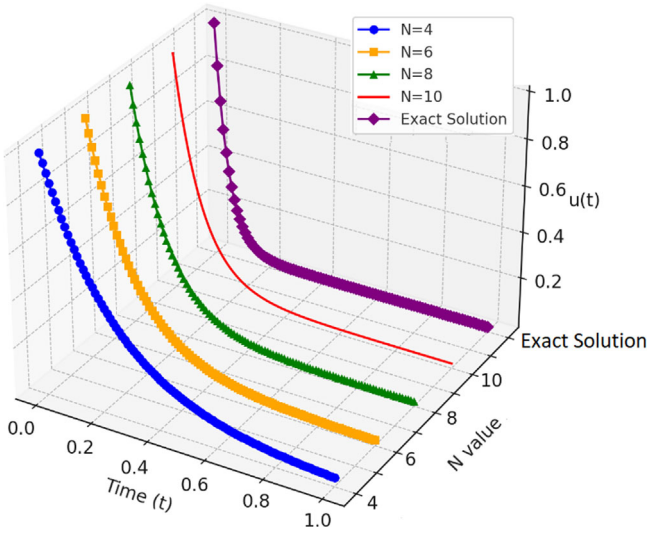


FIGURE 8 | Three-dimensional comparison of approximate and exact solutions over time for increasing polynomial degrees ($N = 4, 6, 8, 10$). (*Source:* Authors' computation). [Colour figure can be viewed at wileyonlinelibrary.com]

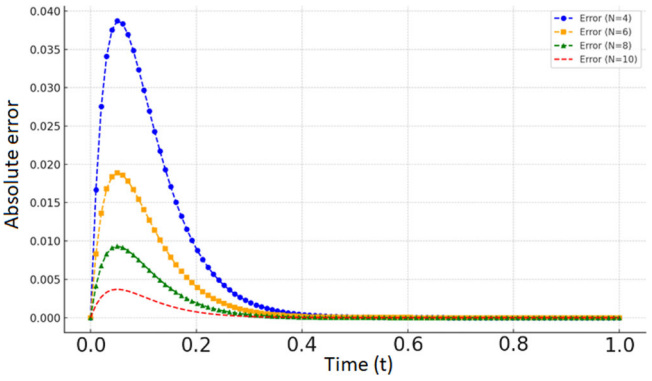


FIGURE 9 | Absolute error distributions for different values of N in solving the Pantograph equation. (*Source:* Authors' computation). [Colour figure can be viewed at wileyonlinelibrary.com]

This convergence behavior is particularly important in control and engineering systems, such as robotic motion planning or delayed signal processing, where approximation quality directly influences system performance. The method's ability to maintain high fidelity even at moderate polynomial degrees (e.g., $N = 6$ or $N = 8$) highlights its computational efficiency and practical applicability to real-time or embedded platforms. Furthermore, the gradual alignment of the solution surfaces underscores the method's robustness against typical instabilities arising from delay-induced stiffness in differential models.

Figure 9 presents the absolute error distributions for various values of N, showcasing the accuracy of the proposed spectral matrix collocation method in solving the nonlinear Pantograph equation. As the polynomial degree increases, the error magnitude diminishes significantly, indicating a strong convergence behavior. The plot clearly demonstrates that higher order approximations (e.g., $N = 6$ or $N = 8$ and $N = 10$) produce solutions that are closely aligned with the exact analytical solution, confirming the method's spectral accuracy and effectiveness.

7 | Conclusion and Future Work

This study presents a novel and unified spectral matrix collocation method based on Lucas polynomials for solving a wide class of nonlinear differential equations, including Bratu-type problems, Riccati equations, and higher order systems with constant and variable delays. The proposed approach effectively addresses the numerical challenges associated with delay terms, nonlinearity, and stiffness by constructing well-conditioned derivative and delay matrices within a common functional basis. The method's performance was validated through several benchmark problems, demonstrating excellent agreement with exact and reference solutions [29, 49], as well as superior accuracy compared to conventional techniques such as the integral collocation method and Adomian decomposition. Error analyses and residual plots confirm the method's spectral convergence, numerical stability, and suitability for high-precision applications. While some limitations remain in handling highly discontinuous or strongly state-dependent delays, the proposed framework offers a robust and efficient alternative for modeling complex dynamic systems in applied mathematics, physics, and engineering.

Future research directions include the following:

- Extending the method to systems with variable or distributed delays, commonly found in biological, supply chain, and communication networks.
- Applying the method to fractional-order, stochastic, or hybrid differential models, where long-term memory and multiscale behavior are prominent.
- Generalizing the approach to multi-dimensional partial differential equations, using domain decomposition or spectral element techniques.
- Integrating the method with artificial intelligence frameworks to develop data-driven surrogate models for real-time prediction, control, and optimization.
- Exploring adaptive refinement strategies, where collocation points are redistributed dynamically based on local error estimates. This would enhance accuracy in regions with rapid changes or discontinuities.

These advancements will further enhance the applicability and flexibility of the Lucas spectral matrix collocation method across diverse fields of science and engineering.

Author Contributions

Nurcan Baykuş Savaşaneril: conceptualization, methodology, writing – original draft, writing – review and editing, formal analysis, supervision, project administration. **Mehmet Çevik:** validation, writing – review and editing, investigation.

Conflicts of Interest

The authors declare no conflicts of interest.

10991476, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/mma.70110 by Nurcan BAYKUS SAVASANERIL - Dokuz Eylul University , Wiley Online Library on [24/09/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Data Availability Statement

All Mathematica codes developed and employed in this study for solving the Bratu, Riccati, and Pantograph-type delay differential equations using the Lucas polynomial-based spectral collocation method are provided in Appendices A–C at the end of this paper. These scripts can be used to reproduce the figures and tables presented in this manuscript.

References

1. A. Mohsen, "A Simple Solution of the Bratu Problem," *Computers and Mathematics With Applications* 67 (2014): 26–33, https://doi.org/10.1016/j.camwa.2013.10.003.

2. A. K. Nasab, Z. P. Atabakan, and Z. P. Kılıçman, "An Efficient Approach for Solving Nonlinear Troesch's and Bratu's Problems by Wavelet Analysis Method," *Mathematical Problems in Engineering* 2013 (2013): 825817, https://doi.org/10.1155/2013/825817.

3. B. Batiha, "Numerical Solution of Bratu-Type Equations by the Variational Iteration Method," *Hacettepe Journal of Mathematics and Statistics* 39 (2010): 23–29.

4. E. Deeba, S. A. Khuri, and S. Xie, "An Algorithm for Solving Boundary Value Problems," *Journal of Computational Physics* 159 (2000): 125–138, https://www.sciencedirect.com/science/article/pii/S0021999100964520.

5. F. Ghomanjani and S. Shateyi, "A New Approach for Solving Bratu's Problem," *Mathematics* 52 (2019): 336–346, https://doi.org/10.1515/dema-2019-0023/html.

6. F. Ghomanjani, "A Numerical Method for Solving Bratu's Problem," *Palestine Journal of Mathematics* 11 (2022): 372–377, https://pjm.ppu.edu/sites/default/files/papers/PJM_May_2022_372_to_377.pdf.

7. H. Q. Kafri and S. A. Khuri, "Bratu's Problem: A Novel Approach Using Fixed-Point Iterations and Green's Functions," *Computer Physics Communications* 198 (2016): 97–104, https://www.sciencedirect.com/science/article/pii/S0010465515003446.

8. H. N. Hassan and M. S. Semary, "Analytic Approximate Solution for the Bratu's Problem by Optimal Homotopy Analysis Method," *Communications in Numerical Analysis* 2013 (2013): 1–14, https://d-nb.info/1077638124/34.

9. H. Caglar, N. Caglar, M. Özer, A. Valarıstos, and A. N. Anagnostopoulos, "B-Spline Method for Solving Bratu's Problem," *International Journal of Computer Mathematics* 87, no. 8 (2009): 1885–1891, https://doi.org/10.1080/00207160802545882.

10. J. Boyd and P. Boyd, "Chebyshev Polynomial Expansions for Simultaneous Approximation of Two Branches of a Function With Application to the One-Dimensional Bratu Equation," *Applied Mathematics and Computation* 143 (2003): 189–200, https://www.sciencedirect.com/science/article/pii/S0096300302003454.

11. M. Al-Mazmumy, A. Al-Mutairi, and K. Al-Zahrani, "An Efficient Decomposition Method for Solving Bratu's Boundary Value Problem," *American Journal of Computational Mathematics* 7 (2017): 84–93, https://doi.org/10.4236/ajcm.2017.71007.

12. M. Baccouch and H. Temimi, "A New Derivation of the Closed-Form Solution of Bratu's Problem," *International Journal of Applied and Computational Mathematics* 9 (2023): 1–8, https://doi.org/10.1007/s40819-023-01570-y.

13. M. Mekonnen, G. Gofe, and H. Garoma, "Numerical Solutions of Second Order Initial Value Problems of Bratu-Type Equations," *International Journal of Engineering & Applied Sciences (IJEAS)* 12 (2020): 15–31, https://dergipark.org.tr/en/pub/ijeas/issue/54727/714965.

14. N. Karamollahi, G. B. Loghmani, and M. Heydari, "A Computational Method to Find Dual Solutions of the One-Dimensional Bratu Problem," *Journal of Computational and Applied Mathematics* 388 (2021): 113309, https://www.sciencedirect.com/science/article/pii/S0377042720306002.

15. R. K. Lodhi, S. F. Aldosary, K. S. Nisar, and A. Alsaadi, "Numerical Solution of Non-Linear Bratu-Type Boundary Value Problems via Quintic B-Spline Collocation Method," *AIMS Mathematics* 7 (2022): 7257–7273, https://doi.org/10.3934/math.2022405.

16. S. A. Khuri, "A New Approach to Bratu's Problem," *Applied Mathematics and Computation* 147 (2004): 1131–1136, https://doi.org/10.1016/S0096-3003(02)00656-2.

17. S. Abbasbandy, M. S. Hashemi, and C. S. Liu, "The Lie-Group Shooting Method for Solving the Bratu Equation," *Communications in Nonlinear Science and Numerical Simulation* 16 (2011): 4238–4249, https://doi.org/10.1016/j.cnsns.2011.03.033.

18. U. Umesh, "Numerical Simulation of Bratu's Problem Using a New Form of the Adomian Decomposition Technique," *International Journal of Numerical Methods for Heat and Fluid Flow* 33 (2023): 2295–2307, https://doi.org/10.1108/HFF-11-2022-0656.

19. M. Çevik, N. B. Savaşaneril, and M. Sezer, "A Review of Polynomial Matrix Collocation Methods in Engineering and Scientific Applications," *Archives of Computational Methods in Engineering* 32 (2025): 1–3373.

20. H. N. Hassan and M. A. E. Tawil, "An Efficient Analytic Approach for Solving Two-Point Nonlinear Boundary Value Problems by Homotopy Analysis Method," *Mathematical Methods in the Applied Sciences* 34 (2011): 977–989, https://doi.org/10.1002/mma.1416.

21. M. A. Akbar, "The Improved *F*-Expansion Method With Riccati Equation and Its Applications in Mathematical Physics," *Cogent Mathematics* 4 (2017): 1282577, https://doi.org/10.1080/23311835.2017.1282577.

22. A. Mısır, "A New Analytic Solution Method for a Class of Generalized Riccati Differential Equations," *Universal Journal of Mathematics and Applications* 6 (2023): 1–6, https://doi.org/10.32323/ujma.1143751.

23. D. Ritelli, "A Forgotten Differential Equation Studied by Jacopo Riccati Revisited in Terms of Lie Symmetries," *Mathematics* 9, no. 11 (2007): 1312, https://www.mdpi.com/2227-7390/9/11/1312.

24. T. Harko, F. S. N. Lobo, and M. K. Mak, "Analytical Solutions of the Riccati Equation With Coefficients Satisfying Integral or Differential Conditions With Arbitrary Functions," *Universal Journal of Applied Mathematics* 2 (2014): 109–118, https://www.hrpub.org/download/20140105/UJAM6-12601983.pdf.

25. X. Li, "The Improved Riccati Equation Method and Exact Solutions to mZK Equation," *International Journal of Differential Equations* (2012): 596762, https://www.hindawi.com/journals/ijde/2012/596762/.

26. M. S. Islam, K. Khan, and M. A. Akbar, "Application of the Improved *F*-Expansion Method With Riccati Equation to Find the Exact Solution of the Nonlinear Evolution Equations," *Journal of the Egyptian Mathematical Society* 25 (2017): 13–18, https://www.sciencedirect.com/science/article/pii/S1110256X16300293.

27. M. R. Ali, M. A. Khattab, and S. M. Mabrouk, "Investigation of Traveling Wave Solutions for the (3+1)-Dimensional Hyperbolic Nonlinear Schrödinger Equation Using Riccati Equation and F-Expansion Techniques," *Optical and Quantum Electronics* 55 (2023): 1–16, https://doi.org/10.1007/s11082-023-05236-3.

28. M. A. E. Abdelrahman and M. A. Sohaly, "The Riccati-Bernoulli Sub-ODE Technique for Solving the Deterministic (Stochastic) Generalized-Zakharov System," *International Journal of Mathematics and Systems Science* 4 (2021): 810, https://systems.enpress-publisher.com/index.php/IJMSS/article/view/810.

29. N. Nikolaev, "Exact Solutions for the Singularly Perturbed Riccati Equation and Exact Wkb Analysis," *Nagoya Mathematical Journal* 250 (2023): 434–469, https://doi.org/10.1017/nmj.2022.38.

30. A. Ardjouni and A. Djoudi, "Fixed Points and Stability in Linear Neutral Differential Equations With Variable Delays," *Nonlinear Analysis* 74 (2011): 2062–2070, https://doi.org/10.1016/j.na.2010.10.050.

10991476, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/mma.70110 by Nurcan BAYKUS SAVASANERIL - Dokuz Eylul University , Wiley Online Library on [24/09/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

31. B. Benhammouda and H. Vazquez-Leal, "A New Multi-Step Technique With Differential Transform Method for Analytical Solution of Some Nonlinear Variable Delay Differential Equations," *Springer Plus* 5 (2016): 1723, https://doi.org/10.1186/s40064-016-3386-8.

32. B. Benhammouda, H. Vazquez-Leal, and L. Hernandez-Martinez, "Procedure for Exact Solutions of Nonlinear Pantograph Delay Differential Equations," *British Journal of Mathematics & Computer Science* 40, no. 19 (2014): 2738–2751, https://doi.org/10.9734/BJMCS/2014/11839.

33. B. Căruntu and C. Bota, "Analytical Approximate Solutions for A General Class of Non-Linear Delay Differential Equations," *Scientific World Journal* 2014 (2014): 631416, https://doi.org/10.1155/2014/631416.

34. B. Dorociakova and R. Olach, "Some Notes to Existence and Stability of the Positive Periodic Solutions for a Delayed Nonlinear Differential Equations," *Open Mathematics* 14 (2016): 361–369, https://doi.org/10.1515/math-2016-0033.

35. C. Jin and J. Luo, "Fixed Points and Stability in Neutral Differential Equations With Variable Delay," *Proceedings of the American Mathematical Society* 136 (2008): 909–918.

36. G. Chen, L. Dingshi, O. V. Gaans, and S. V. Lunel, "Stability of Nonlinear Neutral Delay Differential Equations With Variable Delays," *Electronic Journal of Differential Equations* 118 (2017): 1–14, https://doaj.org/article/c933cbf4f8044b49a6e4ee6911850915.

37. S. Gümgüm, N. Baykus-Savasaneril, Ö. K. Kürkçü, and M. Sezer, "A Numerical Technique Based on Lucas Polynomials Together With Standard and Chebyshev-Lobatto Collocation Points for Solving Functional Integro-Differential Equations Involving Variable Delays," *Sakarya University Journal of Science* 22, no. 6 (2018): 1659–1668, https://dergipark.org.tr/en/pub/saufenbilder/issue/31266/384592.

38. S. Gümgüm, N. Baykus Savasaneril, Ö. K. Kürkçü, and M. Sezer, "Lucas Polynomial Solution of Nonlinear Differential Equations With Variable Delays," *Hacettepe Journal of Mathematics and Statistics* 49, no. 6 (2020): 20191–20212, https://doi.org/10.15672/hujms.460975.

39. I. Ali, H. Brunner, and T. Tang, "A Spectral Method for Pantograph-Type Delay Differential Equations and Its Convergence Analysis," *Journal of Computational Mathematics* 27, no. 2–3 (2009): 254–265, https://www.math.hkbu.edu.hk/~ttang/Papers/JCM272-254.pdf.

40. J. G. Dix, "Asymptotic Behavior of Solutions to a First-Order Differential Equation With Variable Delays," *Computers & Mathematics With Applications* 50 (2005): 1791–1800, https://www.sciencedirect.com/science/article/pii/S0898122105004244.

41. L. B. Cocom, A. G. Estrella, and E. A. Vales, "Solving Delay Differential Systems With History Functions by the Adomian Decomposition Method," *Applied Mathematics and Computation* 218 (2012): 5994–6011, https://www.sciencedirect.com/science/article/pii/S0096300311014287.

42. L. Ding, X. Li, and Z. Li, "Fixed Points and Stability in Nonlinear Equations With Variable Delays, Fixed Point Theory and Applications," 14pp (2010), https://fixedpointtheoryandalgorithms.springeropen.com/articles/10.1155/2010/195916.

43. M. M. Khader, "Numerical and Theoretical Treatment for Solving Linear and Nonlinear Delay Differential Equations Using Variational Iteration Method," *Arab Journal of Mathematical Sciences* 19, no. 2 (2013): 243–256, https://www.sciencedirect.com/science/article/pii/S1319516612000321.

44. M. Cetin, M. Sezer, and C. Guler, "Lucas Polynomial Approach for System of High-Order Linear Differential Equations and Residual Error Estimation," *Mathematical Problems in Engineering* 2015 (2015): 625984, https://doi.org/10.1155/2015/625984.

45. N. B. Savaşaneril and M. Sezer, "Hybrid Taylor-Lucas Collocation Method for Numerical Solution of High-Order Pantograph Type Delay Differential Equations With Variables Delays," *Applied Mathematics and Information Sciences* 11, no. 6 (2017): 1795–1801, https://www.naturalspublishing.com/Article.asp?ArtcID=13792.

46. S. Davaeifar and J. Rashidinia, "Solution of a System of Delay Differential Equations of Multipantograph Type," *Journal of Taibah University for Science* 11 (2017): 1141–1157, https://doi.org/10.1016/j.jtusci.2017.03.005.

47. W. M. Abd-Elhameed and Y. H. Youssri, "Spectral Solutions for Fractional Differential Equations via a Novel Lucas Operational Matrix of Fractional Derivatives," *Romanian Journal of Physics* 61, no. 5–6 (2016): 795–813, https://rjp.nipne.ro/2016_61_5-6/RomJPhys.61.p795.pdf.

48. W. M. Abd-Elhameed and Y. H. Youssri, "Generalized Lucas Polynomial Sequence Approach for Fractional Differential Equations," *Nonlinear Dynamics* 89 (2017): 1341–1355, https://doi.org/10.1007/s11071-017-3519-9.

49. S. Gumgum, N. B. Savaşaneril, O. K. Kurkcu, and M. Sezer, "A Numerical Technique Based on Lucas Polynomials Together With Standard and Chebyshev-Lobatto Collocation Points for Solving Functional Integro-Differential Equations Involving Variable Delays," *Sakarya University Journal of Science* 22, no. 6 (2018): 1659–1668, https://doi.org/10.16984/saufenbilder.384592.

50. M. Shams, N. Kausar, P. Agarwal, and M. A. Shah, "Triangular Intuitionistic Fuzzy Linear System of Equations With Applications: An Analytical Approach," *Applied Mathematics in Science and Engineering* 32 (2024): 2299385, https://doi.org/10.1080/27690911.2023.2299385.

51. M. Shams, N. Kausar, N. Yaqoob, N. Arif, and G. M. Addis, "Techniques for Finding Analytical Solution of Generalized Fuzzy Differential Equations With Applications," *Complexity* 2023 (2023): 1–31, https://doi.org/10.1155/2023/3000653.

52. M. Shams, N. Kausar, K. Alayyash, M. M. al-Shamiri, N. Arif, and R. Ismail, "Semi-Analytical Scheme for Solving Intuitionistic Fuzzy System of Differential Equations," *IEEE Access* 11 (2023): 33205–33223, https://doi.org/10.1109/ACCESS.2023.3241482.

53. M. Shams, N. Kausar, D. Pamucar, E. Özbilge, and B. Tantay, "Modified Block Homotopy Perturbation Method for Solving Triangular Linear Diophantine Fuzzy System of Equations," *Advances in Mechanical Engineering* 15, no. 3 (2023): 1–17, https://doi.org/10.1177/16878132231159519.

54. M. Shams, N. Kausar, D. Pamucar, E. Özbilge, and B. Tantay, "Computationally Semi-Numerical Technique for Solving System of Intuitionistic Fuzzy Differential Equations With Engineering Applications," *Advances in Mechanical Engineering* 14, no. 12 (2022): 1–17, https://doi.org/10.1177/16878132221142128.

55. S. K. Elagan and M. Sayed, "On the Exact Solutions of Fractional Differentional Equations Using Improved Riccati Equations Method," *International Journal of Scientific and Engineering Research* 5 (2014): 795–811, https://www.ijser.org/researchpaper/On-the-exact-solutions-of-fractional-differentional-equations.pdf.

56. K. Erdem, S. Yalçınbaş, and M. Sezer, "A Bernoulli Polynomial Approach With Residual Correction for Solving Mixed Linear Fredholm Integro Differential-Difference Equations," *Journal of Difference Equations and Applications* 19 (2013): 1619–1631, https://doi.org/10.1080/10236198.2013.768619.

10991476, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/mma.70110 by Nurcan BAYKUS SAVASANERIL - Dokuz Eylul University , Wiley Online Library on [24/09/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Appendix A

Lucas Polynomial Approximation for Bratu Equation

The following Mathematica code provides an implementation of the Lucas polynomial-based collocation method for solving the Bratu equation:

```
(* Lucas polynomials definition *)
ClearAll[Lucas]
Lucas[0, x_] := 2
Lucas[1, x_] := x
Lucas[n_, x_] := x Lucas[n - 1, x] - Lucas[n - 2, x]

(* Approximate solution using degree N=3 *)
Nmax = 3;
a = Array[a, Nmax + 1, 0];
u[x_] := Sum[a[i] Lucas[i, x], {i, 0, Nmax}];

(* Bratu residual function *)
lambda = 1;
residual[x_] := D[u[x], {x, 2}] + lambda Exp[u[x]];

(* Boundary conditions: u(0)=0 and u(1)=0 solved manually *)
eq0 = u[0] == 0;
eq1 = u[1] == 0;
bcSol = Solve[{eq0, eq1}, {a[0], a[1]}][[1]];
uFixed[x_] := u[x] /. bcSol;

(* Collocation points *)
collocationPoints = {0.25, 0.5, 0.75};
residualFixed[x_] := D[uFixed[x], {x, 2}] + lambda Exp[uFixed[x]];
equations = Table[residualFixed[xi] == 0, {xi, collocationPoints}];

(* Solve for a[2], a[3] *)
sol = FindRoot[equations, {{a[2], 0.1}, {a[3], 0.1}}];

(* Approximate solution *)
uApprox[x_] := uFixed[x] /. sol;

(* Plot the result *)
Plot[uApprox[x], {x, 0, 1}, PlotStyle -> Red,
PlotLabel -> "Lucas Approximation to Bratu",
AxesLabel -> {"x", "u(x)"}, PlotRange -> All]
```

Appendix B

Lucas Polynomial Approximation for Riccati Equation

The following Mathematica code shows how the Lucas polynomial-based method can be applied to the Riccati differential equation using collocation:

```
(* Lucas polynomial definition *)
ClearAll[Lucas]
Lucas[0, x_] := 2
Lucas[1, x_] := x
Lucas[n_, x_] := x Lucas[n - 1, x] - Lucas[n - 2, x]

(* Degree of approximation *)
Nmax = 3;
a = Array[a, Nmax + 1, 0];
u[x_] := Sum[a[i] Lucas[i, x], {i, 0, Nmax}];

(* Riccati residual: u' + u^2 = 1 *)
residual[x_] := D[u[x], x] + u[x]^2 - 1;

(* Enforcing boundary condition u(0)=0 *)
a0Fix = a[0] -> -Sum[a[i] Lucas[i, 0], {i, 1, Nmax}]/2;
uFixed[x_] := u[x] /. a0Fix;

(* Updated residual *)
residualFixed[x_] := D[uFixed[x], x] + uFixed[x]^2 - 1;

(* Collocation and solution *)
collocationPoints = {0.25, 0.5, 0.75};
equations = Table[residualFixed[xi] == 0, {xi, collocationPoints}];
sol = FindRoot[equations, {{a[1], 0.1}, {a[2], 0.1}, {a[3], 0.1}}];

(* Final approximate solution *)
uApprox[x_] := uFixed[x] /. sol;

(* Plot the result *)
Plot[uApprox[x], {x, 0, 1}, PlotStyle -> Blue,
PlotLabel -> "Lucas Approximation to Riccati Equation",
AxesLabel -> {"x", "u(x)"}, PlotRange -> All]
```

Appendix C

Lucas Polynomial Approximation for a Delay Differential Equation

This Mathematica code demonstrates the Lucas polynomial-based collocation method applied to a second-order delay differential equation of pantograph type:

```
(* Lucas polynomial definition *)
ClearAll[Lucas]
Lucas[0, x_] := 2
Lucas[1, x_] := x
Lucas[n_, x_] := x Lucas[n - 1, x] - Lucas[n - 2, x]

(* Parameter *)
alpha = 0.5;

(* Degree of approximation *)
Nmax = 3;
a = Array[a, Nmax + 1, 0];
u[x_] := Sum[a[i] Lucas[i, x], {i, 0, Nmax}];

(* Delay differential equation residual: u"(x) + u(α x) = 0 *)
residual[x_] := D[u[x], {x, 2}] + u[alpha x];

(* Enforcing boundary conditions u(0)=0 and u(1)=0 *)
eq0 = u[0] == 0;
eq1 = u[1] == 0;
bcSol = Solve[{eq0, eq1}, {a[0], a[1]}][[1]];
uFixed[x_] := u[x] /. bcSol;

(* Updated residual with fixed boundary *)
residualFixed[x_] := D[uFixed[x], {x, 2}] + uFixed[alpha x];

(* Collocation and solving for unknowns *)
collocationPoints = {0.25, 0.5, 0.75};
equations = Table[residualFixed[xi] == 0, {xi, collocationPoints}];
sol = FindRoot[equations, {{a[2], 0.1}, {a[3], 0.1}}];

(* Approximate solution *)
uApprox[x_] := uFixed[x] /. sol;

(* Plot *)
Plot[uApprox[x], {x, 0, 1}, PlotStyle -> Purple,
 PlotLabel -> "Lucas Approximation to Pantograph Equation",
 AxesLabel -> {"x", "u(x)"}, PlotRange -> All]
```


10991476, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/mma.70110 by Nurcan BAYKUS SAYASANERIL - Dokuz Eylul University , Wiley Online Library on [24/09/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License