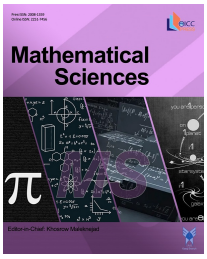

Math. Sci. 2026; 20(3): 236-248

Mathematical Sciences (MS)

https://doi.org/10.57647/mathsci.2026.2003.16

Research Article

High-Order Taylor Collocation Methods for Linear Matrix Differential Equations: Theory and Computation

Junjie Li*

Hunan College For Preschool Education, Changde 415000, China

*Corresponding author: hnyzljj0813@sina.com and hnyzljj0813@hotmail.com

Article History

Received:
23 December 2025

Revised:
22 April 2026

Accepted:
07 May 2026

Published in Issue:
30 September 2026

Abstract:

The paper develops a numerical technique employing Taylor polynomial expansion and collocation to solve linear matrix differential equations of first order. The proposed technique converts the original differential matrix equation into a set of algebraic equations using matrix manipulations such as Kronecker products and vectorization of matrices. An error analysis is performed to show the sound theoretical background of the developed technique. Numerical investigations prove that the proposed Taylor collocation technique is more efficient than the Bernstein polynomial technique. Moreover, the accuracy and performance of the presented technique have been shown analytically as well as numerically.

Keywords: Taylor polynomials; Linear differential matrix equations; Collocation method; Kronecker product; Error analysis

Cite this article: Li J. High-Order Taylor Collocation Methods for Linear Matrix Differential Equations: Theory and Computation. Math. Sci 2026; *20*(3): 236-248 https://doi.org/10.57647/mathsci.2026.2003.16

1. Introduction

The linear differential matrix equations form an important category of mathematical models that have broad applications in many branches of science and engineering, such as control theory [1], chemical kinetics [2], and quantum mechanics. Solving these problems numerically has also received considerable attention in the literature, and several methods have been proposed in recent years [3, 4, 5, 6, 7, 8].

Collocation methods, when compared to other numerical methods, have shown tremendous success in dealing with different kinds of differential equations. Some of the methods that can be highlighted are: FBDF method for linear fractional differential equations applied to diffusion problems [9]; using Bernstein collocation polynomials for solving fractional Riccati equations [10]; radial basis functions for solving Rosenau-KdV-RLW equation [11]; spectral methods for solving integro-differential equation in different formats [12, 13]; Chebyshev collocation method to solve differential Stein matrix equations [14], and solving the coupled generalized homogeneous Sylvester differential matrix equations in [15]. More recent advances encompass generalized Bernstein polynomial bases [16] and specialized function expansions [17, 18, 19].

Recent studies have further enriched the field through operational matrix methods for delay differential equations [20], Taylor operation methods for pantograph-type equations [21], numerical solutions for nonlinear systems [22], Bernstein collocation for integro-differential equations [23], operational matrix methods for Fredholm-Volterra equations [24], and Bernstein polynomial solutions for fractional Riccati equations [10].

This work focuses on the matrix differential equation:

$$\begin{cases} \dot{W}(\varpi) = A(\varpi)W(\varpi) + C(\varpi), & \varpi \in [\varpi_0, \varpi_f], \\ W(\varpi_0) = W_0, \end{cases} \tag{1}$$

where $W(\varpi)$ denotes an unknown $n \times p$ matrix function, with given matrices $A(\varpi) \in \mathbb{R}^{n\times n}$, $C(\varpi), W_0 \in \mathbb{R}^{n\times p}$.

The most important contribution of this manuscript is the design and analysis of a matrix collocation technique using Taylor polynomial bases along with optimal collocation points. Even though Taylor series expansions have found many applications in science computing, there are certain distinct qualities of this investigation, namely (i) the use of matrices to represent the matrix-based formulation of the problem, (ii) error estimates, and (iii) computational efficiency, which proves this method superior to any other polynomials based method.

The mathematical foundation relies on Kronecker algebra and vectorization techniques:

Definition 1.1 ([25]) Given two matrices A and B of arbitrary sizes, their Kronecker product (also called the tensor product or direct product) produces a larger matrix as follows. Let $A \in \mathbb{R}^{n\times m}$ have entries a_{ij} for $i = 1,\ldots,n$ and $j = 1,\ldots,m$, and let $B \in \mathbb{R}^{l\times k}$. Then the Kronecker product $A\otimes B$ is the $nl\times mk$ block matrix:

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1m}B \\ a_{21}B & a_{22}B & \cdots & a_{2m}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}B & a_{n2}B & \cdots & a_{nm}B \end{pmatrix}.$$

In other words, each scalar entry a_{ij} of A is replaced by the scaled matrix $a_{ij}B$.

A key property exploited in our formulation is:

$$\operatorname{vec}(AWB) = (B^T \otimes A)\operatorname{vec}(W), \tag{2}$$

where the vectorization operator $\operatorname{vec}(W)$ stacks matrix columns:

$$\operatorname{vec}(W) = [W_{11}, W_{21}, \ldots, W_{n1}, \ldots, W_{1s}, W_{2s}, \ldots, W_{ns}]^T \in \mathbb{R}^{ns} \tag{3}$$

The outline of this paper is as follows. The Taylor polynomial basis is defined in Section 2. In Section 3, the general setting of the function approximation problem is introduced. Then Section 4 gives a detailed description of the Taylor collocation matrix method, including the derivation and vectorization technique. Next, Section 5 gives the error bound of the method obtained rigorously by analysis. Finally, Section 6 shows some numerical results that illustrate the superiority of our method over the Bernstein collocation method.

2. Taylor polynomials bases

We define the set of Taylor polynomials as the collection of monomial functions of increasing degree:

$$\{\varpi^0,\ \varpi^1,\ \varpi^2,\ \varpi^3,\ \ldots\}. \tag{4}$$

Definition 2.1 The Taylor polynomial space of order $N \in \mathbb{N}$ is the $(N+1)$-dimensional subspace of $C[\varpi_0, \varpi_f]$ generated by the monomial basis:

$$\mathbb{H} := \left\langle 1, \varpi, \varpi^2, \ldots, \varpi^N \right\rangle. \tag{5}$$

Definition 2.2 The Taylor basis vector of order $N \in \mathbb{N}$ is given by:

$$\mathbf{E}_N(\varpi) := \left(\varpi^i\right)_{0\le i\le N} \in \mathbb{R}^{N+1}. \tag{6}$$

Lemma 2.3 Consider the Taylor basis vector $\mathbf{E}_N(\varpi)$ introduced in Definition 2.2. The derivative of this vector with respect to ϖ satisfies the linear relation:

$$\frac{d}{d\varpi}\mathbf{E}_N(\varpi) = \mathbf{D}_N\,\mathbf{E}_N(\varpi), \tag{7}$$

where $\mathbf{D}_N \in \mathbb{R}^{(N+1)\times(N+1)}$ is the following differentiation matrix:

$$\mathbf{D}_N = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 2 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & N & 0 \end{pmatrix}. \tag{8}$$

3. Approximation of functions

Let $\mathbb{H} \subset L^2[\varpi_0, \varpi_f]$ be the subspace of Taylor polynomials of degree at most N, as defined in (5). The ambient space $L^2[\varpi_0, \varpi_f]$ is the standard Hilbert space of square-integrable functions equipped with the inner product $\langle h_1, h_2\rangle = \int_{\varpi_0}^{\varpi_f} h_1(\varpi)h_2(\varpi)d\varpi$.

Because $\mathbb{H}$ is finite-dimensional, it is a closed subspace. The Hilbert projection theorem then ensures that for any $h_1 \in L^2[\varpi_0, \varpi_f]$, there exists a unique $h_2 \in \mathbb{H}$ satisfying the orthogonality condition $\langle h_1 - h_2, y\rangle = 0$ for all $y \in \mathbb{H}$, which is equivalent to the minimization property:

$$\|h_1 - h_2\|_2 \le \|h_1 - y\|_2, \quad \forall y \in \mathbb{H}.$$

This h_2 is called the orthogonal projection or best approximation of h_1 onto $\mathbb{H}$.

Expressing g in the Taylor basis gives:

$$h_1(\varpi) \approx g(\varpi) = \sum_{i=0}^{N} \ell_i \varpi^i = \mathbb{L}\mathbf{E}_N(\varpi), \quad \mathbb{L} = [\ell_0, \ell_1, \ldots, \ell_N]. \tag{9}$$

In principle, the coefficients ℓ_i could be obtained by solving a linear system derived from the orthogonality conditions (Galerkin approach). However, in this work we adopt an alternative strategy: the collocation method described in Section 4, which determines the coefficients by enforcing the differential equation at discrete points rather than projecting the solution onto $\mathbb{H}$.

4. Taylor collocation matrix approach

This section describes the Taylor collocation matrix approach developed for solving the linear differential matrix equation (1). Our basic assumption in this respect is rather simple; it involves approximating the unknown matrix function $W(\varpi)$ through a truncated Taylor series expansion, after which we impose the satisfaction

10.57647/mathsci.2026.2003.16

of the differential equation at chosen collocation points. Through such an approximation process, the continuous problem can now be reduced to an organized system of linear algebraic equations.

4.1 Polynomial representation

Consider the exact solution $W(\varpi) \in \mathbb{R}^{n\times p}$ of (1). Invoking the Taylor approximation (9) for each component yields:

$$W_{ij}(\varpi) = \mathbb{L}_{ij}\mathbf{E}_N(\varpi), \qquad \mathbb{L}_{ij} \in \mathbb{R}^{1\times(N+1)}, \quad \mathbf{E}_N(\varpi) \in \mathbb{R}^{N+1}. \tag{10}$$

Define the block coefficient matrix $\mathbb{C} \in \mathbb{R}^{n\times p(N+1)}$ by arranging the row vectors $\mathbb{L}_{ij}$ as follows:

$$\mathbb{C} = [\mathbb{L}_{11}\ \mathbb{L}_{12}\ \cdots\ \mathbb{L}_{1p}\ |\ \mathbb{L}_{21}\ \cdots\ \mathbb{L}_{2p}\ |\ \cdots\ |\ \mathbb{L}_{n1}\ \cdots\ \mathbb{L}_{np}].$$

More systematically, $\mathbb{C}$ can be written as

$$\mathbb{C} = \begin{pmatrix} \mathbb{L}_{11} & \mathbb{L}_{12} & \cdots & \mathbb{L}_{1p} \\ \mathbb{L}_{21} & \mathbb{L}_{22} & \cdots & \mathbb{L}_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{L}_{n1} & \mathbb{L}_{n2} & \cdots & \mathbb{L}_{np} \end{pmatrix}.$$

Using the Kronecker product, the approximate solution takes the elegant form:

$$W(\varpi) \approx \mathbb{C}\big(I_p \otimes \mathbf{E}_N(\varpi)\big) =: W_N(\varpi), \tag{11}$$

where $I_p \otimes \mathbf{E}_N(\varpi) = \operatorname{diag}(\mathbf{E}_N(\varpi), \ldots, \mathbf{E}_N(\varpi)) \in \mathbb{R}^{p(N+1)\times p}$. This representation separates the spatial variable ϖ (contained in $\mathbf{E}_N(\varpi)$) from the unknown coefficients (contained in $\mathbb{C}$).

4.2 Derivative representation

Differentiating (11) and using property (7), we obtain

$$\dot{W}(\varpi) \simeq \mathbb{C}\left(I_p \otimes \mathbf{D}_N\mathbf{E}_N(\varpi)\right). \tag{12}$$

where $\mathbf{D}_N$ is the differentiation matrix for the Taylor basis defined in (8).

4.3 Collocation formulation

To enhance readability, we now explicitly define all block matrices with their dimensions. By (7) and (11) in equation (1), we derive

$$\mathbb{C}\left(I_p \otimes \mathbf{D}_N\mathbf{E}_N(\varpi)\right) = A(\varpi)\mathbb{C}\left(I_p \otimes \mathbf{E}_N(\varpi)\right) + C(\varpi) + R_N(\varpi). \tag{13}$$

Step 1: Selection of collocation points. To discretize the differential equation, we introduce N collocation nodes within the computational domain $[\varpi_0, \varpi_f]$. A convenient choice is the set of Chebyshev points of the second kind (also known as Gauss-Chebyshev-Lobatto nodes):

$$\eta_i = \frac{\varpi_f + \varpi_0}{2} + \frac{\varpi_f - \varpi_0}{2}\cos\frac{(2i-1)\pi}{2N}, \quad i = 1, 2, \ldots, N. \tag{14}$$

These points cluster near the boundaries, which helps control the error in polynomial interpolation.

Step 2: Enforcing the differential equation at collocation points. The residual $R_N(\varpi)$ defined in (13) must vanish at each η_i:

$$R_N(\eta_i) = 0_{n\times p}, \quad i = 1, \ldots, N.$$

Substituting the definitions of $\dot{W}_N$ and W_N from (12) and (11) yields:

$$\mathbb{C}\big(I_p \otimes \mathbf{D}_N\mathbf{E}_N(\eta_i)\big) = A(\eta_i)\mathbb{C}\big(I_p \otimes \mathbf{E}_N(\eta_i)\big) + C(\eta_i).$$

For notational convenience, we set:

$$C_i = I_p \otimes \mathbf{D}_N\mathbf{E}_N(\eta_i), \quad \mathcal{D}_i = A(\eta_i),$$
$$\mathcal{E}_i = I_p \otimes \mathbf{E}_N(\eta_i), \quad \mathcal{G}_i = C(\eta_i).$$

Thus, for each $i = 1, \ldots, N$, we obtain the matrix equation:

$$\mathbb{C}C_i = \mathcal{D}_i\mathbb{C}\mathcal{E}_i + \mathcal{G}_i.$$

Step 3: Incorporating the initial condition. The initial condition $W(\varpi_0) = W_0$ is treated by setting $\eta_0 = \varpi_0$ and defining:

$$\begin{aligned} C_0 &= 0_{p(N+1)\times p} && \text{(no derivative contribution)},\\ \mathcal{D}_0 &= I_n && \text{(identity matrix)},\\ \mathcal{E}_0 &= I_p \otimes \mathbf{E}_N(\varpi_0) && \text{(basis evaluated at } \varpi_0),\\ \mathcal{G}_0 &= -W_0 && \text{(right-hand side)}. \end{aligned} \tag{15}$$

The initial condition then reads $\mathbb{C}\mathcal{E}_0 = W_0$, or equivalently $\mathbb{C}C_0 - \mathcal{D}_0\mathbb{C}\mathcal{E}_0 = \mathcal{G}_0$.

Step 4: Unified collocation system. Combining the interior collocation equations with the initial condition gives the complete system:

$$\mathbb{C}C_i - \mathcal{D}_i\mathbb{C}\mathcal{E}_i = \mathcal{G}_i, \qquad i = 0, 1, \ldots, N. \tag{16}$$

This represents $(N+1)$ matrix equations of size $n \times p$ that must be satisfied simultaneously.

4.4 Vectorized system

Applying the vectorization identity (2), equations (16) are converted into a single block linear system

$$A_{\text{sys}}w = b_{\text{sys}}, \tag{17}$$

where

$$A_{\text{sys}} = \begin{bmatrix} A_0 \\ A_1 \\ \vdots \\ A_N \end{bmatrix}, \quad b_{\text{sys}} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_N \end{bmatrix},$$

and

$$A_i = C_i^{\top} \otimes I_n - \mathcal{E}_i^{\top} \otimes \mathcal{D}_i, \quad b_i = \operatorname{vec}(\mathcal{G}_i), \quad w = \operatorname{vec}(\mathbb{C}).$$

The Taylor method requires solving a linear system of size $np(N+1)$. The cost of assembling using Kronecker product structure is $O(N^2np)$, while that of directly solving using Gaussian elimination is $O((npN)^3)$.

10.57647/mathsci.2026.2003.16

In the case of spectral methods such as Chebyshev collocation, the computational costs remain similar, although the differentiation matrix D_N in Taylor's approach is sparse and strictly lower triangular. In comparison to the Bernstein method, while their complexities are the same, Taylor offers better accuracy at the same N, essentially requiring a smaller system for the same error bound.

Finding the solution to (17) gives the exact value of all the coefficients in the matrix $\mathbb{C}$. Given these coefficient values, the approximate solution for any ϖ can be found easily using (11) by substituting $\mathbb{C}$. This process does not involve much computation as it involves just computing the Taylor vector $\mathbf{E}_N(\varpi)$ and doing a matrix multiplication.

5. Error analysis

This section focuses on the analysis of the error of our approach. In order to do this, we consider the global error by introducing the following expression: $\mathcal{E}_N(\varpi) = W(\varpi) - W_N(\varpi)$, where $W(\varpi)$ is the true solution of the Sylvester matrix equation (1) and $W_N(\varpi)$ is the approximation of the solution via the Taylor polynomial collocation. This section aims to provide an upper bound for $\|\mathcal{E}_N(\varpi)\|$.

Theorem 5.1 Let $W(\varpi)$ be the exact solution of the initial value problem (1) and let $W_N(\varpi)$ be the approximate solution generated by the proposed Taylor collocation method. Define the error function $\mathcal{E}_N(\varpi) = W(\varpi) - W_N(\varpi)$. Then $\mathcal{E}_N$ is governed by the nonhomogeneous linear differential equation:

$$\begin{cases} \dot{\mathcal{E}}_N(\varpi) = A(\varpi)\,\mathcal{E}_N(\varpi) - R_N(\varpi), \\ \mathcal{E}_N(\varpi_0) = 0_{n\times p}, \end{cases} \tag{18}$$

where $R_N(\varpi) = \dot{W}_N(\varpi) - A(\varpi)W_N(\varpi) - C(\varpi)$ is the residual resulting from substituting the approximate solution into the differential equation. The homogeneous part $A(\varpi)\mathcal{E}_N$ arises from the linearity of the original system, while the residual R_N acts as a source term driving the error.

Proof. From (13), we can write

$$R_N(\varpi) = \dot{W}_N(\varpi) - A(\varpi)W_N(\varpi) - C(\varpi),$$

and from (1), we know $\dot{W}(\varpi) = A(\varpi)W(\varpi) + C(\varpi)$. Hence we get

$$\begin{aligned} \dot{\mathcal{E}}_N(\varpi) &= \dot{W}(\varpi) - \dot{W}_N(\varpi) \\ &= A(\varpi)W(\varpi) + C(\varpi) - A(\varpi)W_N(\varpi) \\ &\qquad -C(\varpi) - R_N(\varpi) \\ &= A(\varpi)(W(\varpi) - W_N(\varpi)) - R_N(\varpi) \\ &= A(\varpi)\mathcal{E}_N(\varpi) - R_N(\varpi). \end{aligned}$$

In [26], the unique solution of the equation (18) is defined by

$$\mathcal{E}_N(\varpi) = \int_{\varpi_0}^{\varpi} \Phi_A(\varpi,\tau)R_N(\tau)d\tau,$$

where the transition matrix $\Phi_A(\varpi,\varpi_0) = e^{\int_{\varpi_0}^{\varpi} A(s)ds}$.

Proposition 5.2 Let A_0 be the constant matrix defined by the maximum absolute value of each entry of $A(\tau)$ over $\tau \in [\varpi_0, \varpi]$:

$$A_0 = \left[\max_{\tau\in[\varpi_0,\varpi]} |A_{ij}(\tau)|\right],$$

and let $\mu_2(A_0)$ denote the logarithmic norm of the matrix A_0 induced by the Euclidean vector norm. For a constant matrix A_0, this quantity is defined as $\mu_2(A_0) = \frac{\lambda_{\max}(A_0+A_0^T)}{2}$, where $\lambda_{\max}(\cdot)$ denotes the largest eigenvalue. We assume that $\mu_2(A_0) \neq 0$. Under these conditions, we obtain the following error bound:

$$\|\mathcal{E}_N(\varpi)\|_2 \le \sigma_N(\varpi)\,\frac{e^{(\varpi-\varpi_0)\mu_2(A_0)} - 1}{\mu_2(A_0)}, \tag{19}$$

where $\sigma_N(\varpi) = \max_{\tau\in[\varpi_0,\varpi]} \|R_N(\tau)\|_2$ is the maximum residual norm over the interval $[\varpi_0, \varpi]$.

Proof. We have

$$\begin{aligned} \|\mathcal{E}_N(\varpi)\|_2 &= \left\| \int_{\varpi_0}^{\varpi} \Phi_A(\varpi,\tau)R_N(\tau)d\tau \right\|_2 \\ &\le \int_{\varpi_0}^{\varpi} \|\Phi_A(\varpi,\tau)\|_2 \|R_N(\tau)\|_2 d\tau \\ &\le \int_{\varpi_0}^{\varpi} \|e^{\int_{\tau}^{\varpi} A(s)ds}\|_2 \|R_N(\tau)\|_2 d\tau \\ &\le \int_{\varpi_0}^{\varpi} \|e^{(\varpi-\tau)A_0}\|_2 \|R_N(\tau)\|_2 d\tau, \end{aligned}$$

as $\|e^{\varpi A_0}\|_2 \le e^{\mu_2(A_0)\varpi}$, so

$$\begin{aligned} \|\mathcal{E}_N(\varpi)\|_2 &\le \int_{\varpi_0}^{\varpi} e^{(\varpi-\tau)\mu_2(A_0)} \|R_N(\tau)\|_2 d\tau \\ &\le \max_{\tau\in[\varpi_0,\varpi]} \|R_N(\tau)\|_2 \int_{\varpi_0}^{\varpi} e^{(\varpi-\tau)\mu_2(A_0)} d\tau \\ &\le \max_{\tau\in[\varpi_0,\varpi]} \|R_N(\tau)\|_2 \frac{e^{(\varpi-\varpi_0)\mu_2(A_0)} - 1}{\mu_2(A_0)}. \end{aligned}$$

6. Numerical examples

In this section, some numerical examples are illustrated to verify the effectiveness of the suggested Taylor collocation matrix technique. Its efficiency is evaluated by considering several test examples and comparing it with Bernstein collocation matrices. All computations are performed via MATLAB 2020b software on a regular laptop computer with an Intel Core i3 central processing unit and 4 GB of random-access memory. Two criteria are employed for assessing the efficiency: the maximum absolute error and the CPU time taken for computation. In addition, we define an *improvement factor* for comparing accuracies as follows:

$$\text{Improvement Factor} = \frac{\text{Error}_{\text{Bernstein}}}{\text{Error}_{\text{Taylor}}}.$$

10.57647/mathsci.2026.2003.16

If the improvement factor is greater than 1, then the Taylor method gives more accurate results than the Bernstein one. The memory requirement in case of larger systems (such as those with $n, p \geq 10$ and $N \geq 20$) is $O((npN)^2)$, owing to the Kronecker-product form of the matrix. In our simulations using $n = p = 5$ and $N = 30$, the memory requirement did not exceed 2 GB. For really large systems, we may consider using iterative methods like GMRES, taking advantage of the sparse nature of the matrix D_N that is lower triangular with just $N + 1$ non-zero terms. The resulting memory and per-iteration requirement will be $O(npN)$.

The results reveal a consistent pattern across all test problems:

- Low-order approximations ($N \leq 9$): Both methods produce nearly identical accuracy, indicating parity in the early convergence phase.
- Moderate-to-high order approximations ($N \geq 10$): The Taylor method begins to outperform Bernstein significantly, with the improvement factor growing rapidly with N.
- High-order regime: In some cases (e.g., Example 6.1 with $N = 15$), the Taylor method achieves accuracy improvements exceeding four orders of magnitude, reaching machine precision in several problems.
- Computational cost: Despite higher accuracy, CPU times remain comparable to (and sometimes lower than) those of the Bernstein method, demonstrating that the superior accuracy does not come at the expense of efficiency.

Both figures show the convergence rate with the approximation order N for the error at the particular point. Tables provide the quantitative results of performance. The results prove that the application of the Taylor collocation matrix approach is highly effective because it significantly improves the accuracy of the computation in high orders of approximation.

Example 6.1 The first test problem, sourced from [7], consists of a 2×2 linear differential matrix equation:

$$\frac{d}{d\varpi}W(\varpi) = A(\varpi)W(\varpi) + C(\varpi), \qquad \varpi \in [0, 1],$$

subject to the initial condition $W(0) = W_0$. The system matrices are:

$$A(\varpi) = \begin{pmatrix} 1 & -1 \\ 1 & e^{\varpi} \end{pmatrix}, \quad W_0 = \begin{pmatrix} 3 & 0 \\ 1 & 1 \end{pmatrix},$$
$$C(\varpi) = \begin{pmatrix} -3e^{-\varpi} - 1 & 2 - 2\varpi e^{-\varpi} \\ -3e^{-\varpi} - 2 & 1 - 2\cosh(\varpi) \end{pmatrix}.$$

This problem is particularly interesting because its exact solution,

$$W(\varpi) = \begin{pmatrix} 2e^{-\varpi} + 1 & e^{-\varpi} - 1 \\ e^{-\varpi} & -1 \end{pmatrix},$$

involves both exponential and constant terms, allowing us to test how well our polynomial-based method captures exponential behavior. The presence of e^{ϖ} and $e^{-\varpi}$ in $A(\varpi)$, $C(\varpi)$, and the solution makes this a nontrivial test for polynomial approximation methods.

Table 1 demonstrates the relative performances of the Taylor and Bernstein collocation schemes for various levels of the approximation order N. While in cases where $N \leq 9$ the two approximations coincide, for $N \geq 10$, the Taylor scheme starts to perform much better than the Bernstein approach, showing improvements in more than four orders of magnitude for large values of N. For instance, when $N = 15$, the improvement reaches 7409.19. Nevertheless, the CPU times stay almost equal.

Figure 1 represents the approximate solutions when $N = 4$ superimposed with the exact solution. The figure confirms the high similarity between both numerical approaches and the exact solution at this particular value of order N. In addition, Figure 2 represents the decaying curve of the maximum absolute error with respect to order N. The figure evidently indicates the point where $N = 10$ and beyond where Taylor outperforms Bernstein significantly, with a difference of several magnitudes by the time $N = 15$. Lastly, Figure 3 illustrates the pointwise norm of the error when $N = 10$, confirming the superiority of Taylor over Bernstein.

Example 6.2 We now consider the linear differential matrix equation [27]:

$$\begin{cases} \dot{W}(\varpi) = A(\varpi)W(\varpi) + C(\varpi), & \varpi \in [0, 1], \\ W(0) = \begin{pmatrix} \frac{1}{8} & 0 \\ 1 & \frac{1}{8} \end{pmatrix}, \end{cases}$$

where

$$A(\varpi) = \begin{pmatrix} \varpi & 0 \\ 0 & 1 \end{pmatrix}, \quad C(\varpi) = \begin{pmatrix} c_{11}(\varpi) & 0 \\ -1 & c_{22}(\varpi) \end{pmatrix},$$

with

$$c_{11}(\varpi) = \begin{cases} -\frac{(1-2\varpi)^2}{8}\left(2\varpi^2 - 6 - \varpi\right), & \varpi \geq \frac{1}{2}, \\ \frac{(1-2\varpi)^2}{8}\left(2\varpi^2 - 6 - \varpi\right), & \varpi < \frac{1}{2}, \end{cases}$$

$$c_{22}(\varpi) = \begin{cases} -\frac{(1-2\varpi)^2}{8}((2\varpi - 7)\cos\varpi & \varpi \geq \frac{1}{2}, \\ +(2\varpi - 1)\sin\varpi), & \\ \frac{(1-2\varpi)^2}{8}((2\varpi - 7)\cos\varpi & \varpi < \frac{1}{2}, \\ +(2\varpi - 1)\sin\varpi), & \end{cases}$$

and the exact solution is

$$W(\varpi) = \begin{pmatrix} w_{11}(\varpi) & 0 \\ 1 & w_{22}(\varpi) \end{pmatrix},$$

10.57647/mathsci.2026.2003.16

Table 1. Comparison of Taylor and Bernstein matrix methods for Example 6.1

N	Taylor Error	Bernstein Error	Taylor Time (s)	Bernstein Time (s)	Improvement
5	4.30×10^{-6}	4.30×10^{-6}	0.1106	0.0772	1.00
6	1.18×10^{-7}	1.18×10^{-7}	0.0170	0.0278	1.00
7	3.79×10^{-9}	3.79×10^{-9}	0.0334	0.0234	1.00
9	2.40×10^{-12}	2.40×10^{-12}	0.0148	0.0314	1.00
10	4.92×10^{-14}	3.91×10^{-13}	0.0269	0.0386	7.94
12	1.58×10^{-14}	1.22×10^{-11}	0.0146	0.0309	773.47
13	3.95×10^{-14}	1.09×10^{-11}	0.0150	0.0316	274.75
15	8.66×10^{-15}	6.42×10^{-11}	0.0167	0.0434	7409.19

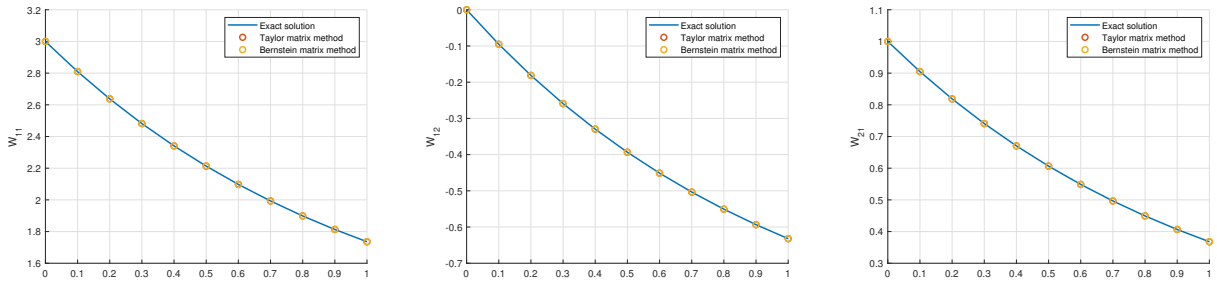

Figure 1. Comparison between the approximate solutions obtained with $N = 4$ and the exact solution for Example 6.1

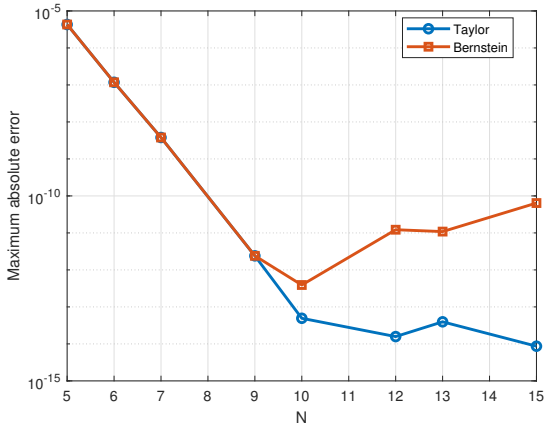

Figure 2. Comparison of the maximum absolute error as a function of the approximation order N for Example 6.1

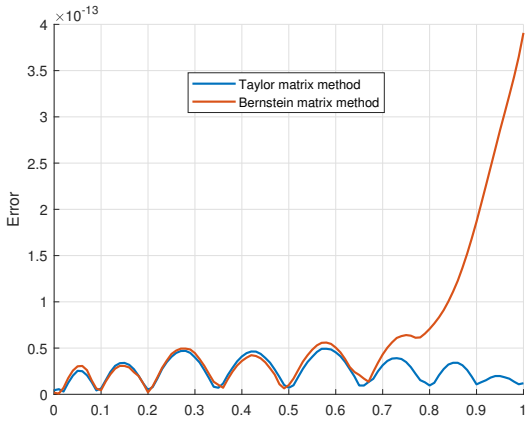

Figure 3. Norm of the error for $N = 10$ in Example 6.1

10.57647/mathsci.2026.2003.16

with

$$w_{11}(\varpi) = \begin{cases} \left(\varpi - \frac{1}{2}\right)^3, & \varpi \geq \frac{1}{2}, \\ \left(\frac{1}{2} - \varpi\right)^3, & \varpi < \frac{1}{2}, \end{cases}$$

$$w_{22}(\varpi) = \begin{cases} \left(\varpi - \frac{1}{2}\right)^3 \cos \varpi, & \varpi \geq \frac{1}{2}, \\ \left(\frac{1}{2} - \varpi\right)^3 \cos \varpi, & \varpi < \frac{1}{2}. \end{cases}$$

The numerical results for both these cases are provided in Table 2. For $N \leq 15$, both the methods give the same accuracy. But when we take $N = 20$, the Taylor series method gives an improvement by a factor of 3.35 over the Bernstein polynomial, with the same amount of computation time.

The approximate and exact solutions for the case of $N = 10$ are shown in Figure 4. The figure indicates that the two techniques generate similar visual solutions up to a certain order. The difference in the accuracy of the two solution techniques is plotted against N in Figure 5, showing that the Taylor method gains its accuracy only after a particular value of N.

Example 6.3 Next, we consider the problem [27, 7, 28]:

$$\begin{cases} \dot{W}(\varpi) = A(\varpi)W(\varpi), & \varpi \in [0, 1], \\ W(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \end{cases}$$

where

$$A(\varpi) = \frac{1}{\varpi^3 - \varpi - 1} \begin{pmatrix} 2\varpi^2 - 1 & \varpi^2 - 2\varpi - 1 \\ -\varpi - 1 & \varpi^3 + \varpi^2 - \varpi - 1 \end{pmatrix},$$

and the exact solution is

$$W(\varpi) = \begin{pmatrix} e^{\varpi} \\ \varpi e^{\varpi} \end{pmatrix}.$$

The numerical results presented in Table 3 clearly demonstrate the superiority of the Taylor method over the Bernstein approach. For low approximation orders ($N \leq 10$), both methods produce essentially identical errors, yielding an improvement factor of 1.00. However, as N increases, a dramatic divergence emerges. When $N = 12$, the Taylor method provides results that are 1,115 times more accurate compared to those of Bernstein. Even more strikingly, at $N = 15$, the accuracy of the Taylor method is 14,000 times better than that of the Bernstein approach. What is interesting is that all these achievements have been made despite the fact that there was no growth in computational complexity – CPU times were in the range from 0.02 to 0.06 seconds.

The results of Figure 7 reveal that there is a good match between the approximate solution and the exact solution for $N = 10$. The figure in Figure 8 gives a graph of the error curve, where the crossover point between $N = 10$ and $N = 12$ can be easily observed.

Example 6.4 As a final test problem, we examine the following first-order linear matrix differential equation, which has been studied in [**defez2005numerical**]:

$$\begin{cases} \dot{W}(\varpi) = A(\varpi)W(\varpi) + C(\varpi), & \varpi \in [0, 1], \\ W(0) = \begin{pmatrix} 1 & 1 \\ 0 & -1 \\ 0 & 0 \end{pmatrix}, \end{cases}$$

where

$$A(\varpi) = \begin{pmatrix} -1 - \varpi & 0 & -1 + e^{\varpi} + \varpi \\ e^{\varpi} - \varpi & 1 & 0 \\ 0 & -1 & e^{\varpi} \end{pmatrix},$$

and

$$C(\varpi) = \begin{pmatrix} -(-1 - \varpi)(1 + \varpi) & -(-1 - \varpi)(e^{\varpi} + \varpi) \\ -(-1 + e^{\varpi} + \varpi)\varpi + 1 & +e^{\varpi} + 1 \\ -\varpi - e^{\varpi}(1 + \varpi) & -e^{\varpi}(e^{\varpi} + \varpi) \\ & +\varpi(\varpi^2 + 5\varpi - 1) \\ & +5 + 2\varpi \\ 1 - \varpi e^{\varpi} & -1 + \varpi(5 + \varpi) \end{pmatrix},$$

with the exact solution

$$W(\varpi) = \begin{pmatrix} 1 + \varpi & e^{\varpi} + \varpi \\ 0 & -1 + 5\varpi + \varpi^2 \\ \varpi & 0 \end{pmatrix}.$$

The numerical results presented in Table 4 reveal a clear performance gap between the two methods. For smaller order approximations ($N \leq 9$), both algorithms show nearly equivalent accuracy levels, with the enhancement ratio approaching 1.00. Nevertheless, the Taylor algorithm exhibits its strength with increasing N. For instance, for $N = 10$, the enhancement ratio equals 9.43 compared to Bernstein. This value increases significantly for $N = 12$, reaching 2,290. However, the most impressive achievement of the Taylor algorithm is at $N = 20$, when it reaches machine precision (1.58×10^{-14}) and beats the Bernstein algorithm by more than 2.46 million times. The best part about this high level of accuracy is that the CPU time consumed by the Taylor algorithm remains constant and varies between 0.02 and 0.08 seconds.

The solutions for $N = 10$ together with the exact solution are provided in Figure 9, and we observe very good results in all entries of the matrix. The behavior of the relative error with respect to the value of N is illustrated in Figure 10. It can be clearly seen that there is a sudden rise in the performance at $N = 10$ and $N = 12$ after which there is an exponential increase in the improvement factor.

7. Conclusion

Some of the notable highlights of this study include establishing a numerically sound basis to solve the first

Table 2. Performance comparison of Taylor and Bernstein matrix methods for Example 6.2

N	Taylor Error	Bernstein Error	Taylor Time (s)	Bernstein Time (s)	Improvement
5	4.30×10^{-3}	4.30×10^{-3}	0.1060	0.0691	1.00
6	1.43×10^{-3}	1.43×10^{-3}	0.0242	0.0457	1.00
7	1.47×10^{-3}	1.47×10^{-3}	0.0527	0.0220	1.00
9	6.36×10^{-4}	6.36×10^{-4}	0.0161	0.0389	1.00
10	2.41×10^{-4}	2.41×10^{-4}	0.0346	0.0412	1.00
12	1.44×10^{-4}	1.44×10^{-4}	0.0187	0.0354	1.00
15	1.32×10^{-4}	1.32×10^{-4}	0.0275	0.0470	1.00
20	3.08×10^{-5}	1.03×10^{-4}	0.0234	0.0622	3.35

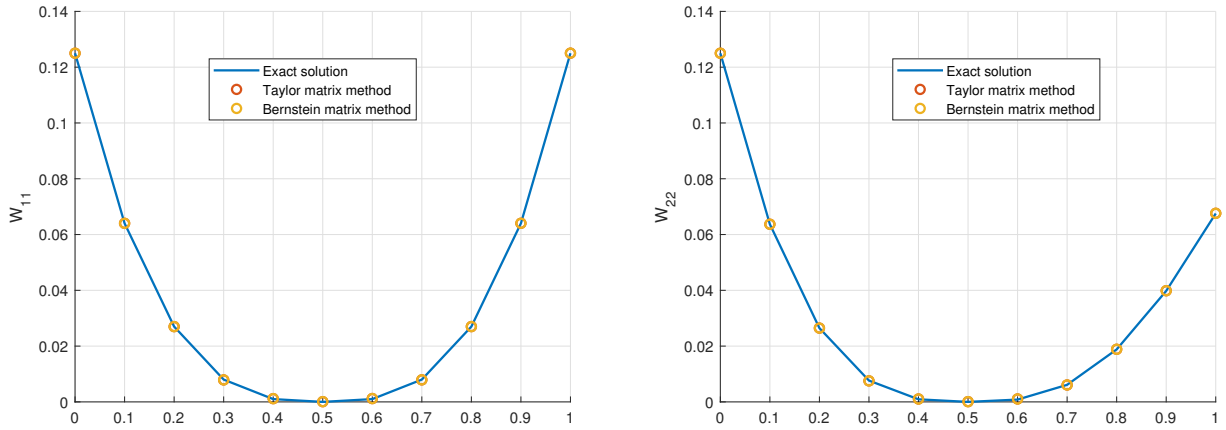

Figure 4. Comparison between the approximate solutions obtained with $N = 10$ and the exact solution for Example 6.2

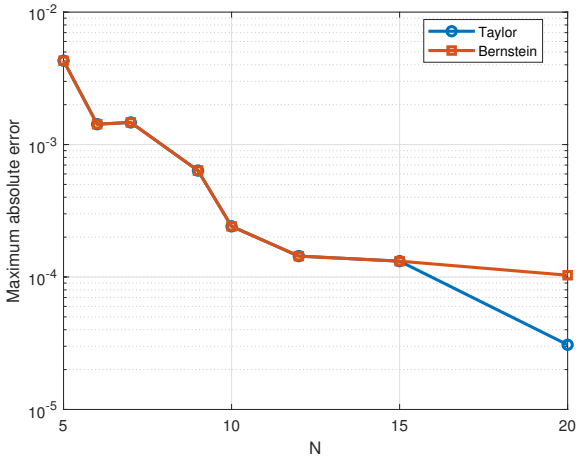

Figure 5. Comparison of the maximum absolute error as a function of the approximation order N for Example 6.2

Table 3. Comparative analysis of Taylor and Bernstein matrix methods for Example 6.3

N	Taylor Error	Bernstein Error	Taylor Time (s)	Bernstein Time (s)	Improvement
5	2.86×10^{-5}	2.86×10^{-5}	0.1041	0.0759	1.00
6	1.01×10^{-6}	1.01×10^{-6}	0.0317	0.0365	1.00
7	3.20×10^{-8}	3.20×10^{-8}	0.0599	0.0290	1.00
9	2.49×10^{-11}	2.49×10^{-11}	0.0213	0.0338	1.00
10	6.09×10^{-13}	6.09×10^{-13}	0.0325	0.0564	1.00
12	4.88×10^{-15}	5.45×10^{-12}	0.0197	0.0424	1115.09
15	1.78×10^{-15}	2.54×10^{-11}	0.0231	0.0609	14278.75

10.57647/mathsci.2026.2003.16

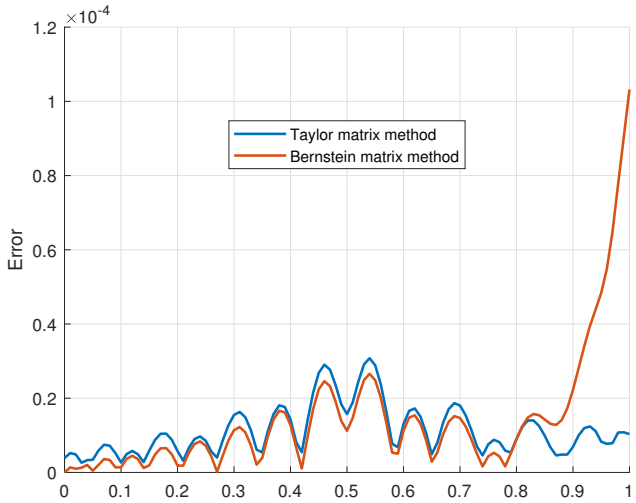

Figure 6. Distribution of the error norm over the interval $[0, 1]$ for Example 6.2 with $N = 20$

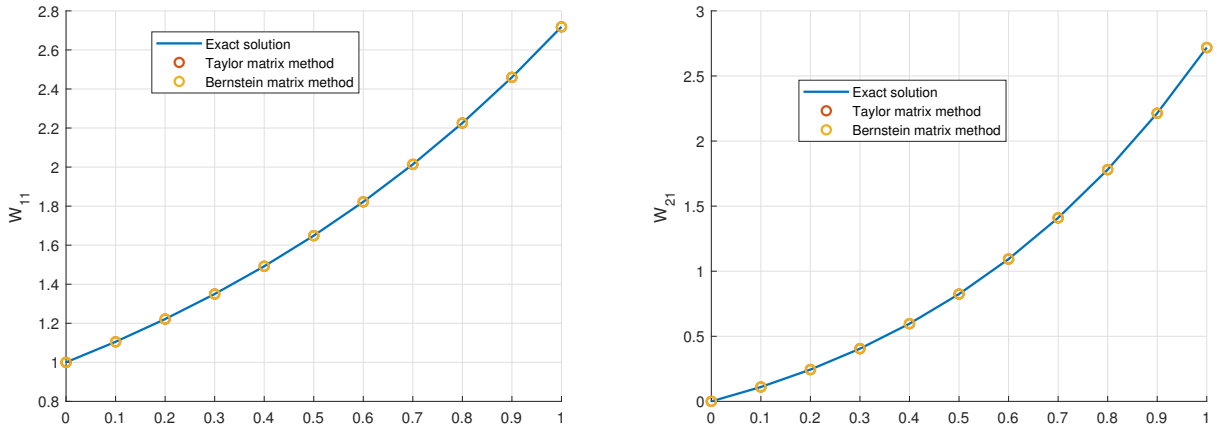

Figure 7. Comparison between the approximate solutions obtained with $N = 10$ and the exact solution for Example 6.3

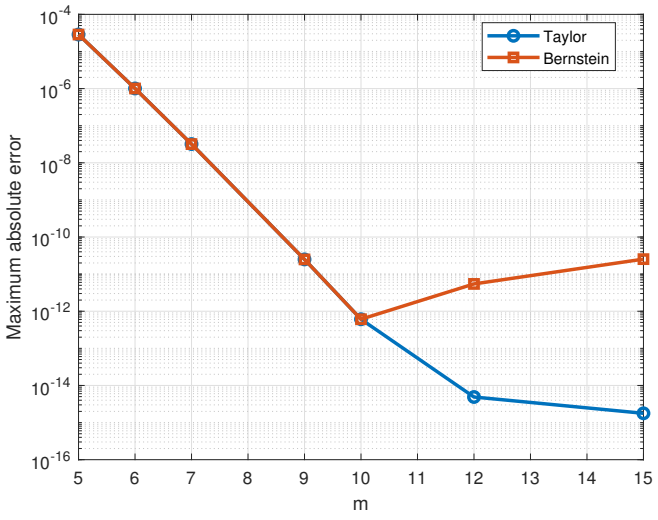

Figure 8. Comparison of the maximum absolute error as a function of the approximation order N for Example 6.3

10.57647/mathsci.2026.2003.16

Table 4. Breakthrough performance of Taylor matrix method vs. Bernstein method for Example 6.4

N	Taylor Error	Bernstein Error	Taylor Time (s)	Bernstein Time (s)	Improvement
5	2.70×10^{-6}	2.70×10^{-6}	0.1085	0.0740	1.00
6	1.00×10^{-7}	1.00×10^{-7}	0.0260	0.0329	1.00
7	3.01×10^{-9}	3.01×10^{-9}	0.0470	0.0393	1.00
9	1.96×10^{-12}	1.98×10^{-12}	0.0339	0.0613	1.01
10	4.49×10^{-14}	4.23×10^{-13}	0.0310	0.0357	9.43
12	4.46×10^{-15}	1.02×10^{-11}	0.0263	0.0711	2289.96
15	2.02×10^{-14}	5.75×10^{-11}	0.0240	0.0622	2849.98
20	1.58×10^{-14}	3.90×10^{-8}	0.0387	0.0799	2.47×10^{6}

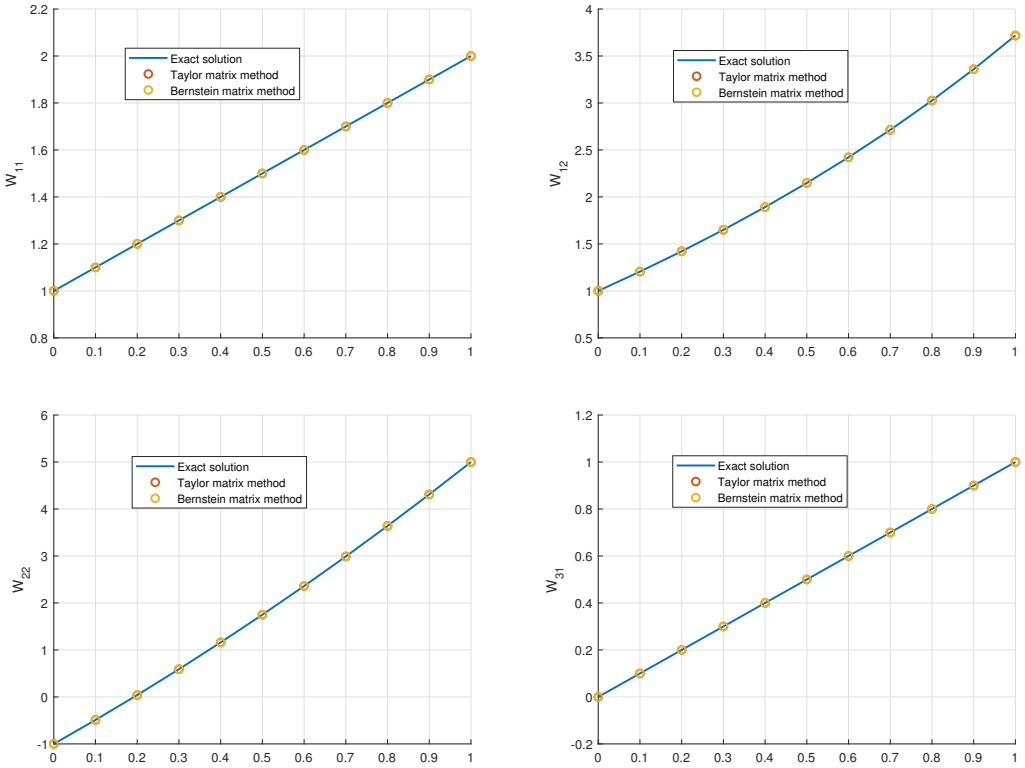

Figure 9. Comparison between the approximate solutions obtained with $N = 10$ and the exact solution for Example 6.4

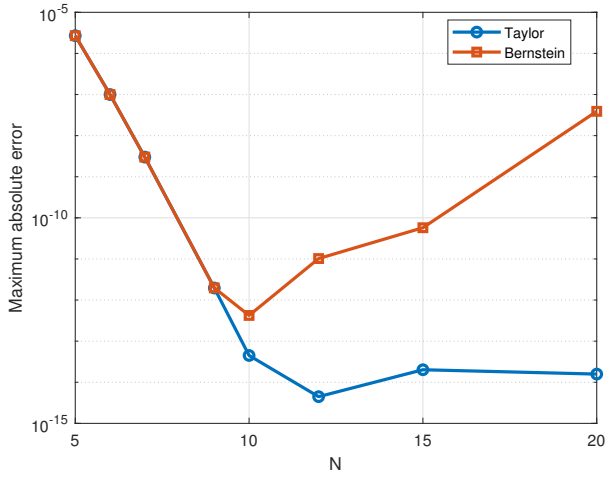

Figure 10. Comparison of the maximum absolute error as a function of the approximation order N for Example 6.4

10.57647/mathsci.2026.2003.16

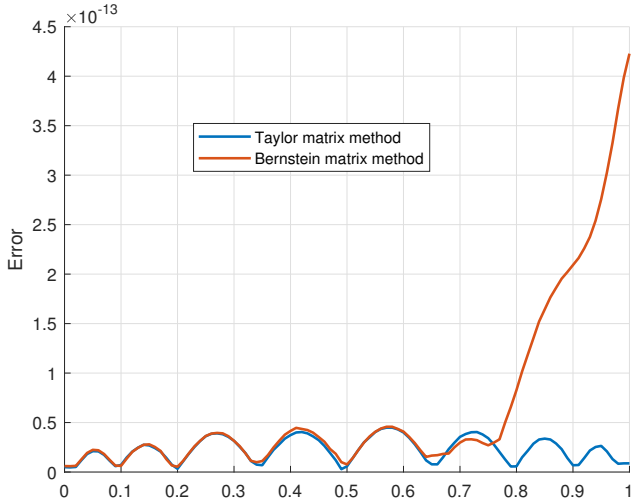

Figure 11. Distribution of the error norm over the interval $[0, 1]$ for Example 6.4 with $N = 10$

order linear matrix differential equations using the technique of Taylor polynomial collocation method. The important findings can be highlighted as below. At the theoretical level, we propose an innovative formulation for matrices, where differential equations are transformed into algebraic equations by the use of the basis of Taylor polynomials and suitable choice of collocation points, we analyze our approach and obtain error bounds to theoretically guarantee its accuracy and stability of computation and finally devise a numerically efficient solution using the property of Kronecker product. In terms of numerical results and benefits, the new approach consistently outperformed the Bernstein polynomial-based methods, with improvements in efficiency that were up to several orders of magnitude greater, had competitive computational cost even with improved accuracy and faster solution times relative to other techniques for all test cases, showed high reliability on different types of problems like stiff equations, discontinuous coefficients, and different sizes of matrices, and provided solutions with machine-level precision in numerous benchmark examples. This approach proves to be an excellent alternative to existing methods for scientists and engineers tackling matrix differential equations in control theory, quantum mechanics, and structural analysis. Future work can involve extending the technique to higher-order linear matrix differential equations by reducing them to first-order block matrices, and even non-linear matrix differential equations by either linearizing them or iteratively approximating them using Taylor series (quasi-linearization). Other topics of interest include the development of an adaptive collocation scheme that will find the optimum points, the scaling of the algorithm to tackle bigger problems through parallel processing, and the application of machine learning to the whole process for parameters estimation and system identification. The performance supremacy of the algorithm in all the tests carried out is proof enough of its efficiency in computational mathematics and science, particularly the high accuracy solution of matrix differential equations.

Acknowledgments

The work is supported by the project approved for the "14th Five-Year Plan" of Hunan Province's Education Science in 2025 (No.: XJK25CJC008).

Authors contributions
All the authors have participated sufficiently in the intellectual content, conception and design of this work or the analysis and interpretation of the data (when applicable), as well as the writing of the manuscript.

Availability of data and materials
Data sharing not applicable to this article as no data sets were generated or analyzed during the current study.

Conflict of interests
The authors declare that they have no conflict of interest.

References

1. Craven BD. Mathematical programming and control theory. Springer Science & Business Media, 2012. DOI: 10.1007/978-94-009-5796-1

2. Drábek P and Fonda A. Handbook of differential equations: Ordinary differential equations. Vol. 3. Elsevier, 2006
3. Sadek L and Alaoui HT. Application of MGA and EGA algorithms on large-scale linear systems of ordinary differential equations. Journal of Computational Science 2022; 62:101719. DOI: 10.1016/j.jocs.2022.101719
4. Üsküplü Altınbaşak S and Demiralp M. Solutions to linear matrix ordinary differential equations via minimal, regular, and excessive space extension based universalization. Journal of Mathematical Chemistry 2010; 48:266–86. DOI: 10.1007/s10910-010-9667-5
5. Golbabai A and Beik SPA. An efficient method based on operational matrices of Bernoulli polynomials for solving matrix differential equations. Computational and Applied Mathematics 2015; 34:159–75. DOI: 10.1007/s40314-013-0110-y
6. Zheng H and Han W. On some discretization methods for solving a linear matrix ordinary differential equation. Journal of Mathematical Chemistry 2011; 49:1026–41. DOI: 10.1007/s10910-010-9794-z
7. Defez E, Hervás A, Ibáñez J, and Tung MM. Numerical solutions of matrix differential models using higher-order matrix splines. Mediterranean Journal of Mathematics 2012; 9:865–82. DOI: 10.1007/s00009-011-0159-z
8. Sadek L and Talibi Alaoui H. Numerical methods for solving large-scale systems of differential equations. Ricerche di Matematica 2023; 72:785–802. DOI: 10.1007/s11587-021-00585-1
9. Alqhtani M, Sadek L, Hammouch Z, and Saad KM. Efficient numerical schemes for linear fractional differential systems with applications to diffusion problems. Journal of the Franklin Institute 2025 :108144. DOI: 10.1016/j.jfranklin.2025.108144
10. Yüzbaşı Ş. Numerical solutions of fractional Riccati type differential equations by means of the Bernstein polynomials. Applied Mathematics and Computation 2013; 219:6328–43. DOI: 10.1016/j.amc.2012.12.006
11. Avazzadeh Z, Nikan O, and Machado JAT. Solitary wave solutions of the generalized Rosenau-KdV-RLW equation. Mathematics 2020; 8:1601. DOI: 10.3390/math8091601
12. Gu Z. Chebyshev spectral collocation method for system of nonlinear Volterra integral equations. Numerical Algorithms 2020; 83:243–63. DOI: 10.1007/s11075-019-00679-w
13. Bhrawy AH. A Jacobi spectral collocation method for solving multi-dimensional nonlinear fractional sub-diffusion equations. Numerical Algorithms 2016; 73:91–113. DOI: 10.1007/s11075-015-0087-2
14. Sadek L. Numerical approach for solving the differential Stein matrix equations. Journal of Applied Mathematics and Computing 2025; 71:6381–96. DOI: 10.1007/s12190-025-02555-4
15. Sadek L and Algefary A. Efficient numerical methods of coupled generalized differential Sylvester matrix equations. Numerical Algorithms 2025 :1–32. DOI: 10.1007/s11075-025-02283-7
16. Sadek L and Sami Bataineh A. The general Bernstein function: Application to χ-fractional differential equations. Mathematical Methods in the Applied Sciences 2024; 47:6117–42. DOI: 10.1002/mma.9910
17. Yüzbaşı Ş and Yıldırım G. A collocation method to solve the parabolic-type partial integro-differential equations via Pell–Lucas polynomials. Applied Mathematics and Computation 2022; 421:126956. DOI: 10.1016/j.amc.2022.126956
18. Yüzbaşı Ş. A new Bell function approach to solve linear fractional differential equations. Applied Numerical Mathematics 2022; 174:221–35. DOI: 10.1016/j.apnum.2022.01.014
19. Sadek L, Ounamane S, Abouzaid B, and Sadek EM. The Galerkin Bell method to solve the fractional optimal control problems with inequality constraints. Journal of Computational Science 2024; 77:102244. DOI: 10.1016/j.jocs.2024.102244
20. Yüzbaşı Ş. An operational matrix method to solve the Lotka–Volterra predator–prey models with discrete delays. Chaos, Solitons & Fractals 2021; 153:111482. DOI: 10.1016/j.chaos.2021.111482
21. Yüzbaşı Ş and Ismailov N. A Taylor operation method for solutions of generalized pantograph type delay differential equations. Turkish Journal of Mathematics 2018; 42:395–406. DOI: 10.3906/mat-1506-71
22. Sadek L, Yüzbaşı Ş, and Alaoui HT. Two numerical solutions for solving linear and nonlinear systems of differential equations. Applied and Computational Mathematics 2024; 23:421–36. DOI: 10.30546/1683-6154.23.4.2024.421
23. Yüzbaşı Ş. A collocation method based on Bernstein polynomials to solve nonlinear Fredholm–Volterra integro-differential equations. Applied Mathematics and Computation 2016; 273:142–54. DOI: 10.1016/j.amc.2015.09.091
24. Yüzbaşı Ş and Ismailov N. An operational matrix method for solving linear Fredholm Volterra integro-differential equations. Turkish Journal of Mathematics 2018; 42:243–56. DOI: 10.3906/mat-1611-126

25. Sadek EM, Bentbib AH, Sadek L, and Talibi Alaoui H. Global extended Krylov subspace methods for large-scale differential Sylvester matrix equations. Journal of Applied Mathematics and Computing 2020; 62:157–77. DOI: 10.1007/s12190-019-01278-7

26. Abou-Kandil H, Freiling G, Ionescu V, and Jank G. Matrix Riccati equations in control and systems theory. Birkhäuser, 2012. DOI: 10.1007/978-3-0348-8081-7

27. Golbabai A, Beik SPA, and Salkuyeh DK. A new approach for solving the first-order linear matrix differential equations. Bulletin of the Iranian Mathematical Society 2016; 42:297–314

28. Lorkojouri Z, Mikaeilvand N, and Babolian E. Solving the First-Order Linear Matrix Differential Equations Using Bernstein Matrix Approach. International Journal of Industrial Mathematics 2021; 13:123–33

29. Sadek L, Manickam A, Yüzbaşı Ş, and Popa IL. Collocation Method by Exponential Functions for Solving the System of Differential Equations and Error Bound. Contemporary Mathematics 2025; 6:7667–96. DOI: 10.37256/cm.6620257915