Iranian Journal of Numerical Analysis and Optimization
Vol. ??, No. ??, ??, pp ??
https://doi.org/??
https://ijnao.um.ac.ir/

Research Article

A novel collocation-based numerical method for higher-order linear ODE systems with pharmacokinetic applications

F. Kaci*

Abstract

This paper introduces a collocation method founded on the independence polynomial of the path graph to solve systems of linear differential equations subject to mixed conditions. The approach converts the original differential system and its associated conditions into an equivalent matrix representation, resulting in a linear algebraic system whose unknowns are the independence coefficients. Several test problems are conducted to assess the numerical efficiency and accuracy of the method. As a practical application, the technique is applied to the three-compartment pharmacokinetic model, illustrating its robustness, numerical stability, and computational simplicity. The results demonstrate the proposed method's effectiveness and reliability in solving complex systems of differential equations.

AMS subject classifications (2020): 65L05; 65L10; 65N35; 05C31

Keywords: Systems of differential equations; Graph; Independence polynomial; Collocation method; Compartment pharmacokinetic model.

*Corresponding author

Received ??? ; revised ??? ; accepted ???

Fatma Kaci
Department of Mathematics, Faculty of Exact Sciences, Mohamed Khider University, Biskra, Algeria. e-mail: fatma.kaci@univ-biskra.dz

How to cite this article
Kaci, F., A novel collocation-based numerical method for higher-order linear ODE systems with pharmacokinetic applications. *Iran. J. Numer. Anal. Optim.*, ??; ??(??): ??-??. ??

1 Introduction

The numerical solution of systems of ordinary differential equations (ODEs) plays an important role in many scientific and technical disciplines, including physics, engineering, biology, and economics [5, 9, 13, 17]. These systems often model complex phenomena involving several interdependent variables and higher-order dynamic relationships. However, their exact solution remains, in the majority of cases, inaccessible, which justifies the use of numerical methods.

Over the past few decades, several numerical approaches have been developed to solve systems of linear differential equations, particularly those of high order. Classical methods include generalized Runge–Kutta methods, finite difference schemes, and spectral methods, as well as wavelet and collocation techniques, which are widely used for their flexibility and accuracy. In particular, spectral, wavelet, and collocation methods rely on polynomial or piecewise polynomial approximations to represent the solution through suitable basis functions. For illustration, Akyüz and Sezer [1] and Ozturk [12] presented Chebyshev collocation methods for solving high-order linear problems, where the original equations are transformed into equivalent matrix forms. In a related study, Çetin, Sezer, and Güler [4] employed Lucas polynomials to solve high-order linear problems with variable coefficients, along with residual error estimation. More recently, Shiralashetti and Kumbinarasaiah [15] introduced a numerical algorithm based on an exact Parseval frame based on Laguerre wavelets. For a class of non-linear Lane–Emden-type equations, Krithikka and Hariharan [10] presented algorithms using Hermite wavelet and Fermat polynomials. Gümgüm, Özdek, and Özaltun [7] proposed a Legendre wavelet method for solving high-order nonlinear equations with variable and proportional delays, while Mall and Chakraverty [11] developed a single-layer Legendre neural network model to tackle nonlinear systems. Basit et al. [3] used a symmetric Bernstein approximation to solve a system of fourth-order nonlinear ODEs on any finite interval $[a, b]$. The technique is based on the operational matrices of Bernstein polynomials using Chebyshev collocation nodes. Although these approaches have shown good performance in several situations, some may encounter difficulties with high-order systems due to issues of numerical stability or implementation complexity.

In this paper, we employ a novel basis function to numerically solve systems of high-order linear differential equations. We propose a collocation method based on independence polynomials of an even path. This basis is derived from graph theory. Although it lacks orthogonality, which can sometimes lead to less favorable numerical conditioning, the practical results indicate that it is nonetheless effective for the problems addressed. A comparison with standard orthogonal polynomials is also provided to highlight the effectiveness and potential of the proposed approach.

A graph $G = (V(G), E(G))$ consists of a finite nonempty set $V(G)$ of n vertices together with a prescribed set $E(G)$ of m unordered pairs of distinct vertices of $V(G)$. Each pair $e = (u, v)$ of vertices in $E(G)$ is an edge of G. We say that the vertices are adjacent if they are joined by an edge. A subset A of $V(G)$ is an *independent set* of G if no two vertices of A are adjacent. The *independence polynomial* of a graph G is the polynomial

$$I(G, x) = \sum_{k=0}^{\alpha(G)} a_k x^k,$$

where a_k is the number of independent set of cardinality k of G and $\alpha(G)$ is the cardinality of the largest independent set of G.

A *path* of order n, denoted P_n, is a graph on n vertices $v_1, \ldots, v_n$ where v_i is adjacent to v_{i+1}, for all $i = 1, 2, \ldots, n-1$. The length of a path is the number of edges in it. We define $P_0 = (\emptyset, \emptyset)$. See Figure 1.

Arocha [2] deduced that the independence polynomial of a path P_n can be computed as follows:

$$I(P_n, x) = \sum_{j=0}^{\lfloor (n+1)/2 \rfloor} \binom{n+1-j}{j} x^j.$$

In particular,

$$I(P_0, x) = 1, \quad I(P_1, x) = 1 + x, \quad I(P_2, x) = 1 + 2x, \quad I(P_3, x) = 1 + 3x + x^2.$$

Figure 1: Paths P_1, P_2, and P_3.

Let us consider the following systems of ordinary differential equations with variable coefficients:

$$\sum_{n=0}^{m} \sum_{j=1}^{k} P_n^{i,j}(x)\, u_j^{(n)}(x) = f_i(x), \quad i = 1, 2, \ldots, k, \tag{1}$$

with the mixed conditions

$$\sum_{n=0}^{m-1} \left(\beta_n^{i,j}\, u_j^{(n)}(a) + \gamma_n^{i,j}\, u_j^{(n)}(b) \right) = \alpha_{i,j}, \quad i = 0, 1, \ldots, m-1, \quad j = 1, 2, \ldots, k. \tag{2}$$

Here, $j = 1, 2, \ldots, k$. The function $u_j^{(0)}(x) = u_j(x)$ is an unknown function and $u \in C^m([a, b])$, while $P_n^{i,j}(x)$ and $f_i(x)$ are known continuous functions defined on the interval $a \leq x \leq b$. The coefficients $\beta_n^{i,j}$, $\gamma_n^{i,j}$, and $\alpha_{n,i}$ are appropriate real constants.

2 Operational matrices for the independence polynomial

The purpose of this paper is to develop an approximate representation of the solution to (1) under conditions (2) by applying the independence polynomial of even paths in the form:

$$u_j(x) \simeq u_{j,N}(x) = \sum_{n=0}^{N} c_{j,n} I(P_{2n}, x), \quad j = 1, 2, \ldots, k, \tag{3}$$

where $c_{j,n}$, $n = 0, \ldots, N$, $j = 1, 2, \ldots, k$, are unknown independence coefficients.

For every $N \in \mathbb{N}$, the independence polynomial vector is given by

$$I_N(x) = [I(P_0, x) \;\; I(P_2, x) \ldots \;\; I(P_l, x)], \qquad N = \lceil \frac{l+1}{2} \rceil.$$

The vector $I_N(x)$ can be written in the matrix form:

$$I_N(x) = X(x)A, \tag{4}$$

where $X(x) = [1 \; x \ldots \; x^N]$ and A is an $(N+1) \times (N+1)$ matrix in which

$$A_{ij} = \begin{cases} \binom{2j-i}{i-1}, & i \leq j, \\ \\ 0, & i > j. \end{cases}$$

Consider the matrix form of the solution $u_{j,N}$:

$$u_{j,N}(x) = I_N(x) \; C_j, \quad C_j = [c_{j,0} \; c_{j,1} \; \ldots \; c_{j,N}]^T \tag{5}$$

and its derivatives $u_{j,N}^{(n)}$:

$$u_{j,N}^{(n)}(x) = I_N^{(n)}(x) C_j, \quad j = 1, \ldots, k, \quad \text{and } n = 1, 2, \ldots, m. \tag{6}$$

From (3), (5), and (6), we obtain

$$u_{j,N}(x) = X(x)\ A\ C_j,$$

$$u_{j,N}^{(n)}(x) = X^{(n)}(x)\ A\ C_j.$$

The matrix $X^{(n)}(x)$ can be found in terms of $X(x)$ as follows:

$$X^{(1)}(x) = X(x)\ L,$$

$$X^{(2)}(x) = X^{(1)}(x)\ L = X(x)\ L^2, \dots$$

$$X^{(n)}(x) = X(x)L^n,$$

where L is the following matrix:

$$L = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \dots & 0 \\ 0 & 0 & 2 & 0 & 0 \dots & 0 \\ 0 & 0 & 0 & 3 & 0 \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \\ 0 \dots & 0 & \dots 0 & 0 & 0 & N \\ 0 \dots & 0 & \dots & 0 & 0 & 0 \end{bmatrix},$$

which gives

$$u_{j,N}^{(n)}(x) = X(x)\ L^n A\ C_j. \tag{7}$$

We also write the following Kronecker product:

$$\mathbf{u}^{(n)}(x) = \mathbf{X}(x)\ \mathbf{L}^n \mathbf{A}\ \mathbf{C}, \tag{8}$$

where

$$\mathbf{u}^{(n)}(x) = \begin{bmatrix} u_{1,N}^{(n)}(x) \\ u_{2,N}^{(n)}(x) \\ \vdots \\ u_{k,N}^{(n)}(x) \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_k \end{bmatrix},$$

and

$$\mathbf{X}(x) = \begin{bmatrix} X(x) & 0 & \dots & 0 \\ 0 & X(x) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & X(x) \end{bmatrix}_{k \times k}, \quad \mathbf{L} = \begin{bmatrix} L & 0 & \dots & 0 \\ 0 & L & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & L \end{bmatrix}_{k \times k},$$

$$\mathbf{A} = \begin{bmatrix} A & 0 & \dots & 0 \\ 0 & A & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A \end{bmatrix}_{k\times k}.$$

The system (1) can consequently be expressed in the matrix form:

$$\sum_{n=0}^{m} \mathbf{P}_n(x)\mathbf{u}^{(n)}(x) = \mathbf{f}(x), \tag{9}$$

where

$$\mathbf{P}_n(x) = \begin{bmatrix} P_n^{1,1}(x) & P_n^{1,2}(x) & \dots & P_n^{1,k}(x) \\ P_n^{2,1}(x) & P_n^{2,2}(x) & \dots & P_n^{2,k}(x) \\ \vdots & \vdots & \ddots & \vdots \\ P_n^{k,1}(x) & P_n^{k,2}(x) & \dots & P_n^{k,k}(x) \end{bmatrix} \quad \text{and} \quad \mathbf{f}(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_k(x) \end{bmatrix}.$$

3 Independence polynomial method (IPM)

To construct the numerical scheme, we introduce the collocation points

$$x_i = a + \frac{b-a}{N}\, i, \quad i = 0, 1, \dots N \text{ and } 0 \le a \le x_i \le b < \infty.$$

Substituting these points into (9) transforms it into a system of matrix equations that forms the basis of our method:

$$\sum_{n=0}^{m} \mathbf{P}_n(x_i)\mathbf{u}^{(n)}(x_i) = \mathbf{f}(x_i), \qquad i = 0, 1, \dots, N. \tag{10}$$

This can be expressed with the following matrix equation:

$$\sum_{n=0}^{m} \mathbf{P}_n\mathbf{U}^{(n)} = \mathbf{F}, \tag{11}$$

where

$$\mathbf{P}_n = \begin{bmatrix} \mathbf{P}_n(x_0) & 0 & \dots & 0 \\ 0 & \mathbf{P}_n(x_1) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{P}_n(x_N) \end{bmatrix},$$

$$\mathbf{F} = \begin{bmatrix} \mathbf{f}(x_0) \\ \mathbf{f}(x_1) \\ \vdots \\ \mathbf{f}(x_N) \end{bmatrix}, \quad \text{and} \quad \mathbf{U}^{(n)} = \begin{bmatrix} \mathbf{u}^{(n)}(x_0) \\ \mathbf{u}^{(n)}(x_1) \\ \vdots \\ \mathbf{u}^{(n)}(x_N) \end{bmatrix}.$$

From (8), we have

$$\mathbf{u}^{(n)}(x_i) = \mathbf{X}(x_i)\mathbf{L}^n\mathbf{A}\mathbf{C},$$

and thus $\mathbf{U}^{(n)}$ can be written in compact form as follows:

$$\mathbf{U}^{(n)} = \tilde{\mathbf{X}}\mathbf{L}^n\mathbf{A}\mathbf{C}, \tag{12}$$

where

$$\tilde{\mathbf{X}} = \begin{bmatrix} \mathbf{X}(x_0) \\ \mathbf{X}(x_1) \\ \vdots \\ \mathbf{X}(x_N) \end{bmatrix}$$

in which

$$\mathbf{X}(x_i) = \begin{bmatrix} X(x_i) & 0 & \dots & 0 \\ 0 & X(x_i) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & X(x_i) \end{bmatrix}_{k\times k}, \qquad i = 0, 1, \dots, N.$$

Substituting the expression given in (12) into (11) leads to the fundamental matrix equation:

$$\left\{ \sum_{n=0}^{m} \mathbf{P}_n\tilde{\mathbf{X}}\mathbf{L}^n\mathbf{A} \right\} \mathbf{C} = \mathbf{F}. \tag{13}$$

The matrices $\mathbf{P}_n, \tilde{\mathbf{X}}$, $\mathbf{L}$, and $\mathbf{A}$ are square block matrices of dimension $k(N+1)$, whereas $\mathbf{C}$ and $\mathbf{F}$ are block vectors of size $k(N+1)$.

Equation (13) can be written as follows:

$$\mathbf{V}\mathbf{C} = \mathbf{F} \text{ or } [\mathbf{V}; \mathbf{F}], \tag{14}$$

where

$$\mathbf{V} = \sum_{n=0}^{m} \mathbf{P}_n\tilde{\mathbf{X}}\mathbf{L}^n\mathbf{A} = [v_{s,t}].$$

Equation (14) gives rise to a linear system containing $k(N+1)$ algebraic equations for the $k(N+1)$ unknown coefficients.

Imposition of conditions

Starting from (8) and the condition (2), the corresponding matrix representation is obtained as follows:

$$\sum_{n=0}^{m-1} \left[\beta_n X(a) + \gamma_n X(b)\right] \mathbf{L}^n \mathbf{A}\ \mathbf{C} = \tilde{\alpha}, \tag{15}$$

where

$$\beta_n = \begin{bmatrix} \beta_n^1 & 0 & \dots & 0 \\ 0 & \beta_n^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \beta_n^k \end{bmatrix}_{k\times k} \quad \text{with} \quad \beta_n^j = \begin{bmatrix} \beta_n^{0,j} \\ \beta_n^{1,j} \\ \vdots \\ \beta_n^{m-1,j} \end{bmatrix}, \qquad j = 1, 2, \dots, k,$$

$$\gamma_n = \begin{bmatrix} \gamma_n^1 & 0 & \dots & 0 \\ 0 & \gamma_n^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \gamma_n^k \end{bmatrix}_{k\times k} \quad \text{with} \quad \gamma_n^j = \begin{bmatrix} \gamma_n^{0,j} \\ \gamma_n^{1,j} \\ \vdots \\ \gamma_n^{m-1,j} \end{bmatrix}, \qquad j = 1, 2, \dots, k,$$

$$\tilde{\alpha} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_k \end{bmatrix}, \quad \text{with} \quad \alpha_j = \begin{bmatrix} \alpha_{0,j} \\ \alpha_{1,j} \\ \vdots \\ \alpha_{m-1,j} \end{bmatrix}, \qquad j = 1, 2, \dots, k.$$

The matrix form (15) can be expressed as

$$\mathbf{WC} = \tilde{\alpha} \text{ or } [\mathbf{W}; \boldsymbol{\alpha}], \tag{16}$$

where

$$\mathbf{W} = \sum_{n=0}^{m-1} \left[\beta_n X(a) + \gamma_n X(b)\right] \mathbf{L}^n \mathbf{A}.$$

Now, we will replace the rows of the augmented matrix $[\mathbf{W}; \boldsymbol{\alpha}]$ by the row mk of the augmented matrix $[\mathbf{V}; \mathbf{F}]$, to get the new augmented matrix

$$\mathbf{\mathbb{V}C} = \mathbb{F} \quad \text{or} \quad [\mathbb{V}; \mathbb{F}]. \tag{17}$$

If $rank\mathbb{V} = rank[\mathbb{V}; \mathbb{F}] = k(N+1)$, then the unknown coefficients vector $\mathbf{C}$ is determined by solving this linear system.

The determined coefficients $c_{j,0}, c_{j,1}, \ldots, c_{j,N}$, for $j = 1, 2, \ldots, k$, are substituted into (3). We get the independence solution:

$$u_{j,N}(x) = \sum_{n=0}^{N} c_{j,n} I(P_{2n}, x), \qquad j = 1, 2, \ldots, k. \tag{18}$$

Error analysis

In this section, we derive an error estimate for the proposed method. The analysis is based on the error bounds previously obtained in [8]. Before stating our main result, we recall the following interpolation theorem.

Theorem 1. [6] Let u be a function in $\mathcal{C}^{N+1}([a, b])$, and let P_N be a polynomial of degree $\leq N$ that interpolates the function u at $(N+1)$ distinct points $x_0, x_1, \ldots, x_N$ in $[a, b]$. Then to each $x \in [a, b]$ there exists a point $\xi_x \in [a, b]$ such that

$$u(x) - P_N(x) = \frac{u^{(N+1)}(\xi_x)}{(N+1)!} \prod_{i=0}^{N} (x - x_i).$$

Let $u = (u_1, u_2, \ldots, u_k)^T$ be the exact solution of (1) and let $P_{j,N}$ denote the interpolating polynomial of u_j for every $j = 1, 2, \ldots, k$. Set $P_N = \big(P_{1,N}, P_{2,N}, \ldots, P_{k,N}\big)^T$.

We can write for every $j = 1, \ldots, k$,

$$u_j(x) = P_{j,N}(x) + R_{j,N}(x),$$

where

$$R_{j,N}(x) = \frac{u_j^{(N+1)}(\xi_x^j)}{(N+1)!} \prod_{i=0}^{N} (x - x_i), \qquad \xi_x^j \in [a, b].$$

Now, let $u_N = \big(u_{1,N}, u_{2,N}, \ldots, u_{k,N}\big)^T$ be the independence solution vector of (1) given by (18); this means that u_N satisfies (2). Therefore u_N and P_N are solutions of $\mathbf{\mathbb{V}C} = \mathbb{F}$ and $\mathbf{\mathbb{V}C}^* = \mathbb{F} + \Delta(\mathbf{F})$, respectively, where $\Delta(\mathbf{F})$ is the $k(N+1)$ vector:

$$\Delta(\mathbf{F}) = \big(\Delta(\mathbf{f}_1), \Delta(\mathbf{f}_2), \ldots, \Delta(\mathbf{f}_k)\big)^T,$$

$$[\Delta(\mathbf{f}_j)]_i = -\sum_{n=0}^{m}\sum_{j=1}^{k} P_n^{i,j}(x_i)\, R_{j,N}^{(n)}(x_i).$$

The following theorem provides an upper bound for the error between the exact solution and the independence solution of the problem (1)–(2).

Theorem 2. Let u and u_N be the exact and independence solution vectors of (1)–(2), respectively, and let P_N be the interpolating polynomial vector of u. Let $\mathbf{V}$,$\mathbf{F}$, $\mathbf{C}$, $\mathbf{C}^*$, and $\Delta(\mathbf{F})$, be defined as above. If each u_j, for every $j = 1, 2, \ldots, k$, is sufficiently smooth, then

$$|u_j(x) - u_{j,N}(x)| \leq |R_{j,N}(x)| + \| X(x)A(C_j - C_j^*) \|_\infty .$$

Proof. We use the term $P_{j,N}$, for every $j = 1, 2, \ldots, k$, as follows:

$$\begin{aligned} |u_j(x) - u_{j,N}(x)| &= |(u_j(x) - P_{j,N}(x)) + (P_{j,N}(x) - u_{j,N}(x))| \\ &\leq |u_j(x) - P_{j,N}(x)| + |u_{j,N}(x) - P_{j,N}(x)| \\ &\leq |R_{j,N}(x)| + |u_{j,N}(x) - P_{j,N}(x)|. \end{aligned}$$

On the other hand, from (7) we have

$$\begin{aligned} |u_{j,N}(x) - P_{j,N}(x)| &= |X(x)AC_j - X(x)A\ C_j^*| \\ &\leq |X(x)A(C_j - C_j^*)| \\ &\leq \|X(x)A(C_j - C_j^*)\|_\infty. \end{aligned}$$

□

4 Numerical examples and discussion

In this paper, several test problems are solved numerically using the proposed method. The numerical results obtained are presented in order to highlight the diversity of the cases treated, in terms of orders, coefficients, types of equations and solutions. This variety illustrates both the generality and the computational efficiency of the proposed method for solving systems of linear ordinary differential equations. Note that all the numerical simulations related to the proposed method have been performed using MATLAB. The absolute errors $|e_{j,N}(x_i)|$ in tables are the values of $|u_j(x_i) - u_{j,N}(x_i)|$ at selected points x_i.

Example 1. Consider the second-order linear differential equation system:

$$\begin{cases} u_1''(x) + xu_1(x) + xu_2(x) = 2, \\ \\ u_2''(x) + 2xu_2(x) + 2xu_1(x) = -2, \end{cases} \qquad 0 \le x \le 1, \tag{19}$$

with the boundary conditions $u_1(0) = 0,\ u_1'(0) = -1,\ u_2(1) = 0$, and $u_2'(1) = -1$. Here, we have

$$\mathbf{P}_0(x) = \begin{bmatrix} x & x \\ 2x & 2x \end{bmatrix}, \quad \mathbf{P}_1(x) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{P}_2(x) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \text{and} \quad \mathbf{f}(x) = \begin{bmatrix} 2 \\ -2 \end{bmatrix}.$$

The set of the collocation points for $N = 2$ is

$$\{x_0 = 0, x_1 = \frac{1}{2}, x_2 = 1\}.$$

Hence, the fundamental matrix equation of the problem from (13) is written as

$$\left\{ \mathbf{P}_0\tilde{\mathbf{X}}\mathbf{L}^0\mathbf{A} + \mathbf{P}_1\tilde{\mathbf{X}}\mathbf{L}^1\mathbf{A} + \mathbf{P}_2\tilde{\mathbf{X}}\mathbf{L}^2\mathbf{A} \right\} \mathbf{C} = \mathbf{F},$$

$$\mathbf{P}_0 = \begin{bmatrix} \mathbf{P}_0(0) & 0 & 0 \\ 0 & \mathbf{P}_0(\frac{1}{2}) & 0 \\ 0 & 0 & \mathbf{P}_0(1) \end{bmatrix}, \quad \mathbf{P}_1 = \begin{bmatrix} \mathbf{P}_1(0) & 0 & 0 \\ 0 & \mathbf{P}_1(\frac{1}{2}) & 0 \\ 0 & 0 & \mathbf{P}_1(1) \end{bmatrix},$$

$$\mathbf{P}_2 = \begin{bmatrix} \mathbf{P}_2(0) & 0 & 0 \\ 0 & \mathbf{P}_2(\frac{1}{2}) & 0 \\ 0 & 0 & \mathbf{P}_2(1) \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \mathbf{f}(0) \\ \mathbf{f}(\frac{1}{2}) \\ \mathbf{f}(1) \end{bmatrix}, \quad \tilde{\mathbf{X}} = \begin{bmatrix} \mathbf{X}(0) \\ \mathbf{X}(\frac{1}{2}) \\ \mathbf{X}(1) \end{bmatrix},$$

$$\mathbf{X}(0) = \begin{bmatrix} X(0) & 0 \\ 0 & X(0) \end{bmatrix}, \quad \mathbf{X}(\frac{1}{2}) = \begin{bmatrix} X(\frac{1}{2}) & 0 \\ 0 & X(\frac{1}{2}) \end{bmatrix},$$

$$\mathbf{X}(1) = \begin{bmatrix} X(1) & 0 \\ 0 & X(1) \end{bmatrix}, \quad \mathbf{L} = \begin{bmatrix} L & 0 \\ 0 & L \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix},$$

$$\mathbf{C} = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}, \text{ where } C_1 = \begin{bmatrix} c_{1,0} \\ c_{1,1} \\ c_{1,2} \end{bmatrix} \text{ and } C_2 = \begin{bmatrix} c_{2,0} \\ c_{2,1} \\ c_{2,2} \end{bmatrix}.$$

Thus, the augmented matrix is calculated as

$$[\mathbf{V};\mathbf{F}] = \begin{bmatrix} 0 & 0 & 6 & 0 & 0 & 0 & ; & 2 \\ 0 & 0 & 0 & 0 & 0 & 6 & ; & -2 \\ \frac{1}{2} & 1 & 63/8 & \frac{1}{2} & 1 & 15/8 & ; & 2 \\ 1 & 2 & 15/4 & 1 & 2 & 39/4 & ; & -2 \\ 1 & 3 & 14 & 1 & 3 & 8 & ; & 2 \\ 2 & 6 & 16 & 2 & 6 & 22 & ; & -2 \end{bmatrix}.$$

On the other hand, from (16), the matrix form of conditions is the following one:

$$[\mathbf{W};\alpha] = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & ; & 0 \\ 0 & 2 & 4 & 0 & 0 & 0 & ; & -1 \\ 0 & 0 & 0 & 1 & 3 & 8 & ; & 0 \\ 0 & 0 & 0 & 0 & 2 & 10 & ; & -1 \end{bmatrix}.$$

The new augmented matrix based on the conditions is given as

$$[\bar{\mathbf{V}};\bar{\mathbf{F}}] = \begin{bmatrix} 0 & 0 & 6 & 0 & 0 & 0 & ; & 2 \\ 0 & 0 & 0 & 0 & 0 & 6 & ; & -2 \\ 1 & 1 & 1 & 0 & 0 & 0 & ; & 0 \\ 0 & 2 & 4 & 0 & 0 & 0 & ; & -1 \\ 0 & 0 & 0 & 1 & 3 & 8 & ; & 0 \\ 0 & 0 & 0 & 0 & 2 & 10 & ; & -1 \end{bmatrix}.$$

Consequently, the system is solved, yielding the following independence coefficient vector:

$$\mathbf{C} = (\bar{\mathbf{V}})^{-1}\bar{\mathbf{F}} = \begin{bmatrix} \frac{5}{6} & \frac{-7}{6} & \frac{1}{3} & \frac{-5}{6} & \frac{7}{6} & \frac{-1}{3} \end{bmatrix}^T.$$

Finally, the vector C is substituted into (5), then

$$u_{1,N}(x) = I_N(x)\ C_1 \quad \text{and} \quad u_{2,N}(x) = I_N(x)\ C_2,$$

where

$$C_1 = \begin{bmatrix} \frac{5}{6} & \frac{-7}{6} & \frac{1}{3} \end{bmatrix}^T \quad \text{and} \quad C_2 = \begin{bmatrix} \frac{-5}{6} & \frac{7}{6} & \frac{-1}{3} \end{bmatrix}^T.$$

We obtain the approximate solutions as follows:

$$u_{1,N}(x) = x^2 - x \quad \text{and} \quad u_{2,N}(x) = -x^2 + x,$$

which are the exact solutions of the system.

Example 2. Consider the fourth-order linear differential equation system [18]:

$$\begin{cases} u_1^{(4)}(x) - \cos(x)u_2^{(2)}(x) + xu_3^{(1)}(x) - u_1(x) = xe^x + \cos^2(x), \\ u_2^{(4)}(x) + \sin(x)u_1^{(3)}(x) + \cos(x)u_1(x) - \cos(x)u_3(x) = (1-e^x)\cos(x), \\ e^{(-x)}u_3^{(4)}(x) + u_2^{(2)}(x) - \cos(x)u_1^{(1)}(x) + u_2(x) = \sin^2(x), \\ 0 \le x \le 1, \end{cases} \tag{20}$$

with the initial conditions $u_1(0) = 0, u_1^{(1)}(0) = 1, u_1^{(2)}(0) = 0, u_1^{(3)}(0) = -1, u_2(0) = 1, u_2^{(1)}(0) = 0, u_2^{(2)}(0) = -1, u_2^{(3)}(0) = 0, u_3(0) = u_3^{(1)}(0) = u_3^{(2)}(0) = u_3^{(3)}(0) = 1$.

The exact solution of the problem is given by $u_1(x) = \sin(x)$, $u_2(x) = \cos(x)$ and $u_3(x) = e^x$.

As observed in Table 1, the accuracy of the computed solution tends to decrease with increasing N. The same table further compares the performance of the proposed method with that of the exponential approximation method reported in [18], used to solve problem (20).

Example 3. Consider the following system of linear differential equations with variable coefficients [12, 15]:

$$\begin{cases} u_1^{(2)}(x) - u_2^{(1)}(x) + u_3^{(1)}(x) - e^{-x}u_1(x) + e^x u_3(x) = e^x - 2e^{-x} + xe^{-x}, \\ u_2^{(2)}(x) + u_1^{(1)}(x) - u_3^{(1)}(x) - u_1(x) = xe^{-x} - e^{-x}, \qquad 0 \le x \le 1, \\ u_3^{(2)}(x) - u_1^{(1)}(x) + u_2^{(1)}(x) + e^x u_2(x) + u_1(x) = 2e^{-x} - xe^{-x} + x, \end{cases} \tag{21}$$

with initial conditions $u_1(0) = 1, u_2(0) = 0, u_3(0) = 1, u_1'(0) = 1, u_2'(0) = 1, u_3'(0) = -1$ and the exact solution $u_1(x) = e^x, u_2(x) = xe^{-x}$ and $u_3(x) = e^{-x}$.

Table 2 presents a comparison of the absolute errors obtained by the proposed method and the Chebyshev operational method [12]. It can be seen that the current method provides a slightly more accurate approximation with slightly lower errors.

In terms of numerical performance, our method also stands out for its significantly reduced computational time. Indeed, it requires approximately 1 second to produce the results, compared to approximately 7 seconds for the method in [12], as indicated in [15]. This difference highlights

Table 1: Absolute errors for the IPM method and the method used in [18] for Example 2.

x_i	$\lvert e_{1,5}(x_i)\rvert$	$\lvert e_{1,5}(x_i)\rvert$ [18]	$\lvert e_{1,10}(x_i)\rvert$	$\lvert e_{1,10}(x_i)\rvert$ [18]
0	2.8623e-17	5.9999e-12	4.4938e-11	1.0221e-9
0.2	1.6081e-08	3.3672e-5	4.3323e-11	3.6854e-6
0.4	2.7146e-07	5.2758e-4	2.3235e-11	3.8405e-5
0.6	1.0021e-06	1.6690e-3	2.0988e-10	1.4109e-4
0.8	2.2176e-05	1.6112e-3	5.9634e-10	3.5199e-4
1	1.3750e-04	3.8807e-3	1.4827e-09	7.6995e-4
x_i	$\lvert e_{2,5}(x_i)\rvert$	$\lvert e_{2,5}(x_i)\rvert$ [18]	$\lvert e_{2,10}(x_i)\rvert$	$\lvert e_{2,10}(x_i)\rvert$ [18]
0	2.2204e-16	3.3999e-11	1.3980e-10	1.8999e-10
0.2	1.7676e-07	5.3148e-5	2.2176e-10	6.7587e-7
0.4	2.8260e-06	8.1712e-4	3.5811e-10	6.7557e-6
0.6	1.5149e-07	2.6665e-3	5.7150e-10	2.3055e-5
0.8	8.8001e-05	3.5885e-3	8.8584e-10	5.0852e-5
1	5.3442e-04	4.1086e-4	1.3963e-09	8.9223e-5
x_i	$\lvert e_{3,5}(x_i)\rvert$	$\lvert e_{3,5}(x_i)\rvert$ [18]	$\lvert e_{3,10}(x_i)\rvert$	$\lvert e_{3,10}(x_i)\rvert$ [18]
0	0	4.4409e-16	2.3906e-11	9.7000e-9
0.2	1.9495e-07	2.4145e-4	7.4221e-11	3.1086e-5
0.4	3.1353e-06	3.7515e-3	1.6324e-10	3.2342e-4
0.6	1.1942e-06	1.1835e-2	2.9924e-10	1.1828e-3
0.8	1.1694e-04	1.1622e-2	4.8362e-10	2.9217e-3
1	7.2002e-04	2.6565e-2	4.1498e-10	6.3328e-3

the computational efficiency of the proposed method, which manages to improve accuracy while significantly reducing computational costs. These results confirm the value of the developed method, both in terms of accuracy and numerical efficiency.

Example 4. Consider the system of fourth-order linear differential equations with variable coefficients, subject to mixed conditions:

$$\begin{cases} u_1^{(4)}(x) + u_2^{(2)}(x) = xe^x - 4e^{-x} - \sin(x), \\ \qquad\qquad\qquad\qquad\qquad\qquad 0 \le x \le 2, \\ u_2^{(4)}(x) + u_1^{(2)}(x) = \sin(x) + xe^{-x} - 2e^{-x}, \end{cases} \tag{22}$$

with

$$u_1(0) + u_1(2) = 2e^{-2}, \quad u_1^{(1)}(0) + u_1^{(2)}(2) = 1 - e^{-2}, \quad u_1^{(2)}(0) = -2,$$

$$u_1^{(3)}(0) = 3, \quad u_2(0) = 0, \quad u_2^{(1)}(0) = 1, \quad u_2^{(2)}(0) = 0, \quad u_2^{(3)}(0) = -1,$$

and the exact solutions $u_1(x) = xe^{-x},\ u_2(x) = \sin(x)$.

Table 2: Comparison of absolute errors and CPU Times for Example 3

	$\lvert e_{1,7}\rvert$		$\lvert e_{2,7}\rvert$			$\lvert e_{3,7}\rvert$	
x_i	[12] CPU time: 7.04 s	IPM CPU time: 0.08 s	[12] CPU time: 7.04 s	IPM CPU time: 0.08 s	x_i	[12] CPU time: 7.04 s	IPM CPU time: 0.08 s
0	2.00e-12	1.02e-12	4.59e-13	1.22e-12	0	0	4.45e-13
0.2	1.61e-07	1.35e-09	1.00e-07	1.33e-08	0.2	2.51e-08	3.61e-09
0.4	1.52e-06	1.47e-09	1.72e-07	2.69e-08	0.4	6.90e-07	1.12e-08
0.6	1.58e-06	9.59e-09	1.08e0-7	3.77e-08	0.6	9.87e-07	2.20e-08
0.8	1.93e-07	2.44e-08	1.70e-07	4.41e-08	0.8	2.25e-07	3.63e-08
1	1.86e-08	4.98e-07	1.01e-08	1.25e-06	1	2.21e-10	7.18e-08

Table 3: Absolute errors obtained using IPM for Example 4

x_i	$\lvert e_{1,8}(x_i)\rvert$	$\lvert e_{2,8}(x_i)\rvert$	$\lvert e_{1,10}(x_i)\rvert$	$\lvert e_{2,10}(x_i)\rvert$
0	7.7201e-04	2.7756e-15	8.9828e-06	1.4778e-12
0.2	5.8858e-04	1.0296e-09	6.9495e-06	1.3847e-11
0.4	4.0525e-04	1.7632e-08	4.9170e-06	1.5365e-10
0.6	2.2220e-04	7.1867e-08	2.8867e-06	6.3313e-10
0.8	3.9576e-05	1.7061e-07	8.5960e-07	1.7425e-09
1	1.4246e-04	3.0795e-07	1.1629e-06	3.9044e-09
1.2	3.2380e-04	4.4658e-07	3.1796e-06	7.7289e-09
1.4	5.0359e-04	6.4648e-07	5.1891e-06	1.4047e-08
1.6	6.7437e-04	2.1068e-06	7.1729e-06	2.6467e-08
1.8	8.0147e-04	1.0797e-05	8.9045e-06	7.9137e-08
2	7.7201e-04	4.6495e-05	8.9827e-06	3.7759e-07

As reported in Table 3, the errors obtained by IPM for $N = 8$ and $N = 10$ are of order 10^{-k} over the domain, indicating that the method provides a good approximation of the exact solution. The decrease of the error when passing from $N = 8$ to $N = 10$ illustrates the expected improvement in accuracy with larger truncation parameters.

Example 5. Consider the initial value problem:

$$\begin{cases} u_1'(x) + u_2'(x) + u_2(x) = x - e^{-x}, \\ \\ u_1'(x) + 4u_2'(x) + u_1(x) = 1 + 2e^{-x}, \end{cases} \qquad 0 \le x \le 1, \tag{23}$$

with $u_1(0) = 1$, $u_2(0) = 0$, and exact solutions

Table 4: Comparison of absolute errors for Example 5

x_i	$\lvert e_{1,5}(x_i)\rvert$ [1]	$\lvert e_{1,5}(x_i)\rvert$ [14]	$\lvert e_{1,5}(x_i)\rvert$ IPM
0.1	4.510522e-05	6.2951e-05	5.9187e-07
0.2	7.985043e-05	1.5714e-04	1.0110e-06
0.5	9.719089e-05	4.3862e-04	1.0978e-06
0.8	8.006002e-05	8.0076e-04	9.0094e-07
1	1.067677e-04	1.0598e-03	1.3659e-05
x_i	$\lvert e_{2,5}(x_i)\rvert$ [1]	$\lvert e_{2,5}(x_i)\rvert$ [14]	$\lvert e_{2,5}(x_i)\rvert$ IPM
0.1	2.247723e-05	4.5250e-04	2.9327e-07
0.2	3.984701e-05	6.5580e-04	5.0134e-07
0.5	4.890662e-05	6.7891e-04	5.4596e-07
0.8	4.064222e-05	7.8896e-04	4.5116e-07
1	5.390356e-05	1.2735e-03	6.7555e-06

$$u_1(x) = e^{-x} + 3\ e^{x/3} - 3, \quad u_2(x) = -\frac{1}{2}e^{-x} + \frac{3}{2}\ e^{-x/3} + x - 1.$$

Table 4 illustrates a comparison of the absolute errors obtained by the presented method with those by the Chebyshev collocation method [1] and the rational Chebyshev collocation method [14] for $N = 5$. The numerical results show that the proposed method provides a noticeable improvement compared to the other methods considered.

Example 6. We consider the sixth-order boundary-value ordinary differential equation: [16]

$$u^{(6)}(x) + u(x) = 6(2x\cos(x) + 5\sin(x)), \qquad -1 \le x \le 1,$$

with $u(-1) = u(1) = 0$, $u'(-1) = u'(1) = 2\sin(1)$ and $u^{(2)}(-1) = -u^{(2)}(1) = -4\cos(1) - 2\sin(1)$. The exact solution is $u(x) = (x^2 - 1)\sin(x)$.

We define the maximum and mean absolute errors, $E_{\max}$ and E_{mean}, respectively, by

$$E_{\max} = \max_{1\le i\le M} |e_N(x_i)|, \qquad E_{\text{mean}} = \frac{1}{M}\sum_{i=1}^{M} |e_N(x_i)|,$$

where M denotes the number of spatial grid points. Table 5 lists the maximum absolute error and the mean error obtained by IPM for various choices of N. We observe that, for each value of N, the maximum absolute error and the mean error are of the same order of magnitude. This indicates that the numerical error is distributed rather uniformly over the spatial grid, with no significant local peaks.

Table 5: $E_{\max}$ and E_{mean} for Example 6

N	$E_{\max}$	E_{mean}
9	1.2227e-03	5.5342e-04
10	3.6091e-04	1.6416e-04
11	2.6185e-05	1.1938e-05
12	6.4901e-06	2.9727e-06
13	2.8248e-07	1.3046e-07
14	1.8677e-06	6.1177e-07

5 Three-compartment pharmacokinetic models

Consider the three-compartment pharmacokinetic model, commonly used to model the kinetics of a drug within the body. This model divides the body into three interconnected compartments: the central compartment $C1$, the site of drug elimination, and two peripheral compartments $C2$ and $C3$, representing fast and slow exchange tissues, respectively. Drug transfers between compartments are modeled by first-order transfer constants: k_{12}, k_{21} between C_1 and C_2 and k_{13}, k_{31} between C_1 and C_3. The elimination occurs from the central compartment with rate constant k_{10}. The system is governed by the following set of ordinary differential equations:

$$\begin{cases} \dfrac{dC_1}{dt} = -(k_{10} + k_{12} + k_{13})C_1 + k_{21}C_2 + k_{31}C_3, \\ \dfrac{dC_2}{dt} = k_{12}C_1 - k_{21}C_2, \\ \dfrac{dC_3}{dt} = k_{13}C_1 - k_{31}C_3. \end{cases} \tag{24}$$

The analytical solution of this system is, in general, difficult to obtain due to the coupling of the equations and the presence of several parameters. Therefore, the use of numerical methods is essential to effectively approximate the concentrations in the different compartments over time.

We provide the numerical solution of the system (24) for $t > 0$ using IPM with the initial conditions: $C_1(0) = 5.0000\mu g \cdot kg^{-1}$ (Dose), $C_2(0) = 0$ and $C_3(0) = 0$. The parameter values for the model are $k_{12} = 0.105\ min^{-1}$, $k_{21} = 0.064\ min^{-1}$, $k_{13} = 0.022\ min^{-1}$, $k_{31} = 0.0034\ min^{-1}$, $k_{10} = 0.0827\ min^{-1}$.

Figure 2 shows the concentration of the drug in each compartment obtained using the proposed IPM method for $N = 10$ and the standard `ode23` solver, illustrating a close agreement between both approaches.

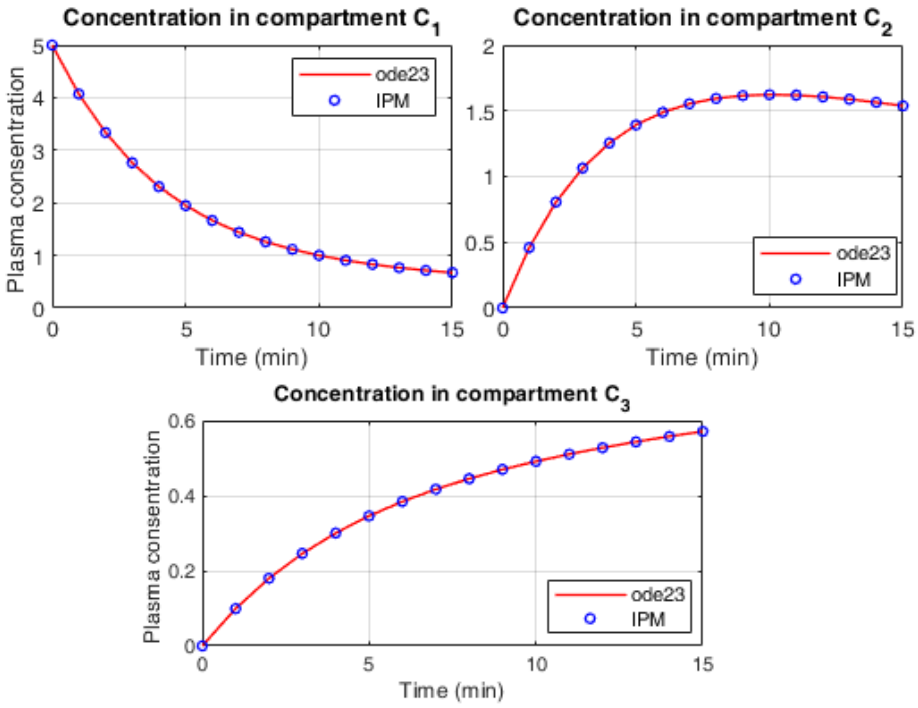

Figure 2: Concentration in the three compartments for $N = 10$

To measure the performance of the proposed approach, we rely on the estimated error function $\tilde{E}_N(x)$, which can also be used when no exact solution is available. If $C_{1,N}(x)$, $C_{2,N}(x)$, and $C_{3,N}(x)$ serve as approximations to the solution of system (24), then inserting these functions and their derivatives into (24) should lead to relations that are satisfied only approximately. For $x_k \in [a, b]$, the associated error terms $\tilde{E}_{i,N}(x_k),\ i = 1, 2, 3$ are defined by

$$\begin{aligned}\tilde{E}_{1,N}(x_k) =& \Big| C'_{1,N}(x_k) + (k_{10} + k_{12} + k_{13})\, C_{1,N}(x_k) - k_{21} C_{2,N}(x_k) \\ & - k_{31} C_{3,N}(x_k) \Big| \approx 0,\end{aligned}$$

$$\tilde{E}_{2,N}(x_k) = \left| C'_{2,N}(x_k) - k_{12} C_{1,N}(x_k) + k_{21} C_{2,N}(x_k) \right| \cong 0,$$

$$\tilde{E}_{3,N}(x_k) = \left| C'_{3,N}(x_k) - k_{13} C_{1,N}(x_k) + k_{31} C_{3,N}(x_k) \right| \cong 0,$$

and $\tilde{E}_{i,N}(x) < 10^{-s_k}$, $i = 1, 2, 3$, is satisfied for a positive constant s_k. If $\max 10^{-s_k} = 10^{-s}$ is specified, then the truncation limit N is increased until $\tilde{E}_{i,N}(x_k)$, $i = 1, 2, 3$, at all points, fall below the prescribed 10^{-s}.

The estimated error functions for $N = 6, 8, 10$ are given in Tables 6, 7, and 8. The graphics of estimated the error functions for $N = 10$ are presented in Figure 3.

Table 6: Estimated error function for $N = 6, 8, 10$ for C_1

x_i	$\tilde{E}_{1,6}$	$\tilde{E}_{1,8}$	$\tilde{E}_{1,10}$
1	3.1036e-04	5.6536e-06	5.6669e-08
3	6.9027e-05	1.1956e-06	5.1010e-12
5	3.4417e-15	5.0910e-07	2.7003e-09
7	3.9963e-05	3.7212e-07	2.2108e-09
9	9.4458e-05	4.4541e-07	4.0115e-10
11	2.2646e-04	8.8744e-07	1.0229e-08
13	6.9269e-04	3.2310e-06	1.1929e-07
15	1.3516e-02	4.6249e-04	9.9699e-06

Table 7: Estimated error function for $N = 6, 8, 10$ for C_2

x_i	$\tilde{E}_{2,6}$	$\tilde{E}_{2,8}$	$\tilde{E}_{2,10}$
1	1.7830e-04	3.2479e-06	3.2555e-08
3	3.9654e-05	6.8684e-07	3.2266e-13
5	1.2670e-14	2.9247e-07	1.5697e-09
7	2.2958e-05	2.1377e-07	1.1885e-09
9	5.4264e-05	2.5588e-07	3.8696e-11
11	1.3009e-04	5.0981e-07	6.6038e-09
13	3.9794e-04	1.8561e-06	6.6827e-08
15	7.7644e-03	2.6569e-04	5.7311e-06

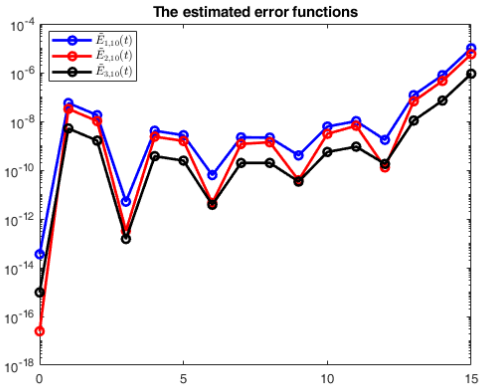

Figure 3: Estimated error functions for $N = 10$

Table 8: Estimated error function for $N = 6, 8, 10$ for C_3

x_i	$\tilde{E}_{3,6}$	$\tilde{E}_{3,8}$	$\tilde{E}_{3,10}$
1	2.8055e-05	5.1107e-07	5.1226e-09
3	6.2397e-06	1.0808e-07	1.5015e-13
5	3.5384e-15	4.6021e-08	2.4530e-10
7	3.6125e-06	3.3638e-08	1.9725e-10
9	8.5386e-06	4.0263e-08	3.3955e-11
11	2.0471e-05	8.0220e-08	9.1691e-10
13	6.2616e-05	2.9207e-07	1.0833e-08
15	1.2218e-03	4.1807e-05	9.0107e-07

6 Conclusion

In this work, a collocation method based on the independence polynomial of the even path has been developed for solving systems of higher-order linear differential equations with mixed conditions. An error analysis was carried out to assess the accuracy of the proposed method, and upper bounds for the approximation error were derived. Furthermore, numerical experiments conducted on various benchmark systems demonstrate the accuracy and efficiency of the proposed method, which often yields superior results compared to existing approaches. Moreover, it was observed that as the parameter N increases, the approximation error decreases, confirming the convergence of the method. Finally, the application of the proposed approach to the three-compartment pharmacokinetic model further validated its robustness and reliability, as the estimated error functions remained very small, indicating excellent numerical performance.

Funding

This research received no external funding.

Conflicts of Interest

The author declares no conflicts of interest.

References

[1] Akyüz, A. and Sezer, M. *Chebyshev polynomial solutions of systems of high-order linear differential equations with variable coefficients*, Appl. Math. Comput., 144(2) (2003), 237–247.

[2] Arocha, J.L. *Propiedades del polinomio independiente de un grafo*, Rev. Cienc. Mat., 5 (1984), 103–110.

[3] Basit, M., Shahnaz, K., Malik, R., Karim, S.A.A. and Khan, F. *An effective approach based on generalized Bernstein basis functions for the system of fourth-order initial value problems for an arbitrary interval*, Mathematics, 11(14) (2023), 3076.

[4] Çetin, M., Sezer, M. and Güler, C. *Lucas polynomial approach for system of high-order linear differential equations and residual error estimation*, Math. Probl. Eng., 2015 (2015), 625984.

[5] Cieślik, S. *Mathematical modeling of the dynamics of linear electrical systems with parallel calculations*, Energies, 14(10) (2021), 2930.

[6] Davis, P.J. and Rabinowitz, P. *Methods of numerical integration*, Academic Press, 2nd ed., 1984.

[7] Gümgüm, S., Özdek, D. and Özaltun, G. *Legendre wavelet solution of high-order nonlinear ordinary delay differential equations*, Turkish J. Math., 43(3) (2019), 1339–1352.

[8] Işık, O.R., Sezer, M. and Güney, M. *A rational approximation based on Bernstein polynomials for high order initial and boundary value problems*, Appl. Math. Comput., 217 (2011), 9438–9450.

[9] Johnston, M.D., Edwards, C.M., Bodmer, W.F., Maini, P.K. and Chapman, S.J. *Mathematical modeling of cell population dynamics in the colonic crypt and in colorectal cancer*, Proc. Natl. Acad. Sci. U.S.A., 104(10) (2007), 4008–4013.

[10] Krithikka, S. and Hariharan, G. *Two robust operational matrix algorithms for solving nonlinear Lane–Emden-type equations in astrophysics using Hermite and Fermat polynomials*, NRIAG J. Astron. Geophys., 14(1) (2025), 1–15.

[11] Mall, S. and Chakraverty, S. *Application of Legendre neural network for solving ordinary differential equations*, Appl. Soft Comput., 43 (2016), 347–356.

[12] Ozturk, Y. *Numerical solution of systems of differential equations using operational matrix method with Chebyshev polynomials*, J. Taibah Univ. Sci., 12(2) (2018), 155–162.

[13] Rafikov, M. and Limeira, E.D.H. *Mathematical modelling of the biological pest control of the sugarcane borer*, Int. J. Comput. Math., 89(3) (2012), 390–401.

[14] Ramadan, M.A., Raslan, K.R. and Nassar, M.A. *An approximate solution of systems of high-order linear differential equations with variable coefficients by means of a rational Chebyshev collocation method*, Electron. J. Math. Anal. Appl., 4(1) (2016), 86–98.

[15] Shiralashetti, S.C. and Kumbinarasaiah, S. *Laguerre wavelets exact Parseval frame-based numerical method for the solution of system of differential equations*, Int. J. Appl. Comput. Math., 6(4) (2020), 1–16.

[16] Siddiqi, S.S. and Twizell, E.H. *Spline solutions of linear sixth order boundary value problems*, Int. J. Comput. Math., 60 (1996), 295–304.

[17] Tzafriri, A.R. *Michaelis–Menten kinetics at high enzyme concentrations*, Bull. Math. Biol., 65(6) (2003), 1111–1129.

[18] Yüzbaşi, Ş. and Sezer, M. *An exponential matrix method for solving systems of linear differential equations*, Math. Methods Appl. Sci., 36(3) (2013), 336–348.