

# Numerical investigation of the solution of higher-order boundary value problems via Euler matrix method

Mohamed El-Gamel<sup>1</sup> · Waleed Adel<sup>1</sup>

Received: 28 August 2017 / Accepted: 30 September 2017 / Published online: 12 October 2017  
© Sociedad Española de Matemática Aplicada 2017

**Abstract** Euler collocation method is developed to approximate the solutions of linear and nonlinear tenth and twelfth-order boundary-value problems. Properties of Euler polynomials and some operational matrices are first presented. These properties are then used to reduce the tenth and twelfth-order boundary-value problems into a system of either linear or nonlinear algebraic equations. Numerical examples illustrate the effectiveness of the method and its possibility of applications to a wide class of problems. The comparison with other method are made. It is shown that Euler collocation method gives better results.

**Keywords** Euler matrix · Higher-order · Euler polynomials · Nonlinear

**Mathematics Subject Classification** Primary 65L60; Secondary 65L10

## 1 Introduction

Twelfth-order boundary-value problems arises in many scientific applications in various branches of science. For example, when heating an infinite horizontal layer of fluid from below along with a uniform magnetic field in the same direction across the fluid and with the fluid is subjected to the action of rotation then, instability will occur. When instability occurs as ordinary convection, it is modelled by tenth-order boundary-value problems or, when instability occurs as over stability, it is modelled by twelfth-order boundary-value problems [5, 6, 8, 9].

Agarwals book [2] contains a detailed theorems which discusses the exitance and uniqueness for solving general higher-order boundary-value problems.

---

✉ Mohamed El-Gamel  
gamel\_eg@yahoo.com

Waleed Adel  
waleedadel85@yahoo.com

<sup>1</sup> Department of Mathematical Sciences, Faculty of Engineering, Mansoura University, Mansoura, Egypt

Several numerical methods have been developed for solving higher-order boundary value problem such as non polynomial spline [26,28], modified Adomian decomposition method [3,31], Legendre operational matrix method [25], Quintic B-spline collocation method [30], iterative method [29], chebychev polynomial solution [10], differential transform technique [24] and variational iteration method [23]. Each of these methods has its essential advantages and disadvantages. An ongoing search for a more effective, general and accurate numerical techniques is a must.

In recent years, a lot of attention has been devoted to the study of Euler method to investigate various scientific models. The efficiency of the method has been formally proved by many researchers [4,16,18–20]. Euler methods for ordinary differential equations has many salient features due to the properties of the basis functions and the manner in which the problem is discretized. The approximating discrete system depends only on parameters of the differential equation.

The aim of this paper is to develop Euler-collocation method for the numerical solution of the following class of linear and non-linear higher-order boundary value problems:

$$u^{(2r)}(x) + \sum_{m=0}^{2r-1} P_m(x)u^{(m)}(x) = \sigma(x, u(x)), \quad 0 \leq x \leq 1, \quad r = 5, 6. \tag{1.1}$$

subject to the boundary conditions

$$u^{(i)}(0) = \alpha_i, \quad u^{(i)}(1) = \beta_i, \quad i = 0, 1, 2, \dots, r - 1, \tag{1.2}$$

where  $\sigma(x, u)$ ,  $P_m(x)$  and  $u(x)$  are continuous functions in  $L^2(0, 1)$ .

The organization of the paper is as follows. In Sect. 2, Euler polynomials along with their relative properties that will be needed later is introduced. In Sect. 3, Euler method is developed for linear and nonlinear higher-order boundary value problems. Error analysis of the method is presented in Sect. 4. In Sect. 5, some numerical examples are presented. Finally, Sect. 6 provides conclusions of the study.

## 2 Preliminaries and fundamentals relations

### 2.1 Euler polynomials

The formulation of Euler numbers and polynomials was first introduced by Euler back in 1740 and like other polynomials they have substantial literature and applications in number theory [7,27,32]. They are well related to Bernoulli polynomials [13,15] where the classical Euler polynomials  $E_n(x)$  can be defined by means of exponential generating functions as

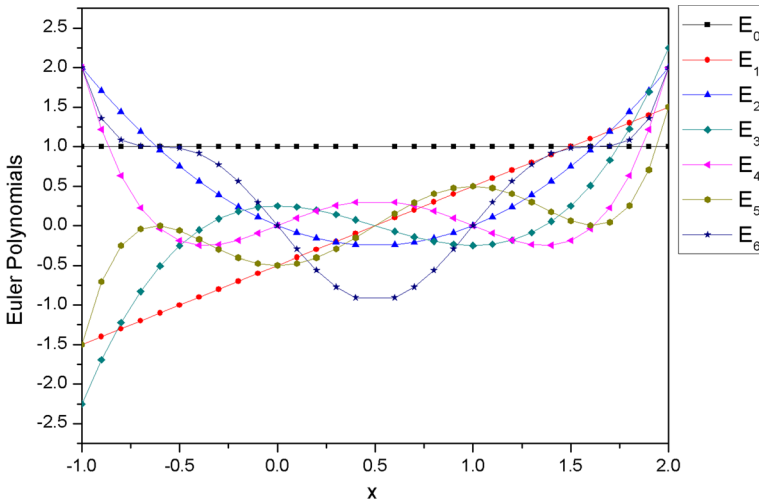
$$\frac{2e^{tx}}{e^t + 1} = \sum_{n=0}^{\infty} E_n(x) \frac{t^n}{n!}, \quad (|t| \leq \pi), \tag{2.1}$$

which is also satisfies the following interesting properties

$$E'_n(x) = n E_{n-1}(x), \quad n = 1, 2, \dots \tag{2.2}$$

$$\sum_{k=0}^n \binom{n}{k} E_k(x) + E_n(x) = 2x^n, \tag{2.3}$$

$$\int_0^1 E_n(x) dx = \frac{-2n!}{(n+1)!} E_{n+1}(0), \tag{2.4}$$



**Fig. 1** Behaviour of Euler polynomials

with  $E_0(x) = 1$  and  $\binom{n}{k}$  is a binomial coefficient. Explicitly, the first few basic Euler polynomials can be expressed by

$$E_1(x) = x - \frac{1}{2}, \quad E_2(x) = x^2 - x, \quad E_3(x) = x^3 - \frac{3}{2}x^2 + \frac{1}{4}.$$

Figure 1 shows the behavior of the first few Euler polynomials on the interval  $[-1, 2]$ .

Also, Euler polynomials  $E_n(x)$  are related to Bernoulli polynomials from the following formula

$$E_n(x) = \frac{1}{n+1} \sum_{k=1}^{n+1} (2 - 2^{k+1}) \binom{n+1}{k} B_k(0) x^{n+1-k}, \quad n = 0, 1, \dots \quad (2.5)$$

where  $B_k(x), k = 0, 1, \dots$  are the Bernoulli polynomials of order  $k$  which satisfies the well known relations [1, 14, 22]

$$B'_n(x) = nB_{n-1}(x), \quad \int_0^1 B_n(x)dx = 0, \quad n = 1, 2, \dots,$$

$$\sum_{k=0}^n \binom{n+1}{k} B_k(x) = (n+1)x^n.$$

For the approximation of any unknown function, Euler basis  $E_n(x)$  has several advantages over other methods since they uses less number of basis functions compared to other methods and their implementations are simple and straightforward. Next, we will introduce the differentiation matrix of these polynomials that will be needed later.

### 2.2 Euler operational matrix of differentiation

Euler polynomials has a lot of interesting relations and properties as mentioned in Sect. 2.1 and the most important relation is the relations of their derivatives which is used for solving

different type of boundary-value problem. We introduce a technique based on Euler approximation to the solution of Eqs. (1.1)–(1.2) expressed in the truncated Euler series form

$$u_N(x) = \sum_{n=0}^N c_n E_n(x) = \mathbf{E}(x)\mathbf{c} \tag{2.6}$$

where  $\{c_n\}_{n=0}^N$  are the unknown Euler coefficients,  $N$  is any chosen positive integer such that  $N \geq 2r$ , and  $E_n(x)$ ,  $n = 0, 1, \dots, N$  are the Euler polynomial of the first kind which are constructed according to Eq. (2.5) and it's relations. Then, the Euler coefficient vector  $\mathbf{c}$  and the Euler vector  $\mathbf{E}(x)$  are given by

$$\mathbf{c}^t = [c_0, c_1, \dots, c_N], \quad \mathbf{E}(x) = [E_0(x), E_1(x), \dots, E_N(x)].$$

According to the expansion in Eq. (2.5) and it's later properties we find

$$\underbrace{\begin{bmatrix} E_0(x) \\ E_1(x) \\ \vdots \\ E_N(x) \end{bmatrix}}_{\mathbf{E}^t(x)} = \underbrace{\begin{bmatrix} \frac{(2-2^2)}{1} \binom{1}{1} B_1(0) & 0 & \dots & 0 \\ \frac{(2-2^3)}{2} \binom{2}{2} B_2(0) & \frac{(2-2^2)}{2} \binom{2}{1} B_1(0) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \frac{(2-2^{N+2})}{N} \binom{N}{N} B_N(0) & \frac{(2-2^{N+1})}{N} \binom{N}{N-1} B_{N-1}(0) & \dots & \frac{(2-2^2)}{N} \binom{N}{1} B_1(0) \end{bmatrix}}_{\mathbf{B}} \underbrace{\begin{bmatrix} 1 \\ x \\ \vdots \\ x^N \end{bmatrix}}_{\mathbf{X}^t(x)}. \tag{2.7}$$

Since  $\mathbf{B}$  is a lower triangular matrix which has a nonzero diagonal elements, it is nonsingular and  $\mathbf{B}^{-1}$  exists. Thus, Euler vector can be described directly from

$$\mathbf{E}(x) = \mathbf{X}(x) \mathbf{B}^t \tag{2.8}$$

note that  $[ ]^t$  denotes transpose of the matrix  $[ ]$  where  $\mathbf{E}^t(x)$  and  $\mathbf{X}^t(x)$  be the  $(N + 1) \times 1$  and  $\mathbf{B}$  is the  $(N + 1) \times (N + 1)$  operational matrix. Now, the matrix form of the solution

$$u_N(x) = \mathbf{X}(x) \mathbf{B}^t \mathbf{c}. \tag{2.9}$$

According to Eq. (2.8) the following formula is concluded evidently. Also, the relation between the matrix  $\mathbf{X}(x)$  and its derivative  $\mathbf{X}^{(1)}(x)$  is

$$\mathbf{X}^{(1)}(x) = \underbrace{\begin{bmatrix} 1, x, x^2, \dots, x^N \end{bmatrix}}_{\mathbf{X}(x)} \underbrace{\begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & N \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}}_{\mathbf{M}}, \tag{2.10}$$

from which we conclude that

$$\mathbf{X}^{(i)}(x) = \mathbf{X}(x) \mathbf{M}^i, \quad i = 0, 1, \dots, 2r \tag{2.11}$$

where  $\mathbf{X}^{(i)}(x)$  is denoting the  $i$ th derivative of  $\mathbf{X}(x)$  then we have

$$u_N^{(i)} = \mathbf{X}(x) \mathbf{M}^i \mathbf{B}^t \mathbf{c}, \quad i = 0, 1, \dots, 2r \tag{2.12}$$

Next, we will illustrate the use of these polynomials along with their operational matrix of derivative for solving Eq. (1.1).

### 3 Application of Euler matrix collocation method

We will illustrate our method based on Eq. (1.1) on two different cases

#### 3.1 Linear high-order BVP

In this case we assign the source term  $\sigma(x, u) = f(x)$  into Eq. (1.1), then the equation becomes

$$u^{(2r)}(x) + \sum_{m=0}^{2r-1} P_m(x)u^{(m)}(x) = f(x), \quad 0 \leq x \leq 1, \quad r = 5, 6. \tag{3.1}$$

Let us seek an approximation of (3.1) expressed in terms of Euler polynomials as

$$u(x) \approx u_N(x) = \sum_{n=0}^N c_n E_n(x) = \mathbf{E}(x) \mathbf{c}$$

where the Euler coefficient vector  $\mathbf{c}$  and the Euler vector  $\mathbf{E}(x)$  are given

$$\mathbf{c}^t = [c_0, c_1, \dots, c_N], \quad \mathbf{E}(x) = [E_0(x), E_1(x), \dots, E_N(x)],$$

then the  $i$ th derivative of  $u_N(x)$  can be expressed in the matrix form by

$$u_N^{(i)}(x) = \mathbf{E}^{(i)} \mathbf{c} = \mathbf{X}(x) \mathbf{M}^i \mathbf{B}^t \mathbf{c} \quad i = 1, 2, \dots, 2r. \tag{3.2}$$

To obtain the approximated solution, we first reduce the terms of Eq. (3.1). The corresponding form of the nonhomogeneous term  $f(x)$  of Eq. (3.1), can be shown

$$f(x) \approx \sum_{n=0}^N f_n E_n(x) = \mathbf{E}(x) \mathbf{F} = \mathbf{X}(x) \mathbf{B}^t \mathbf{F}, \tag{3.3}$$

where  $\mathbf{F} = [f_0, f_1, \dots, f_N]^t$  and  $f_j$ , for  $j = 0, 1, \dots, N$  can be calculated from the a backward linear relation [17]

$$f_N = \frac{1}{N!} \int_0^1 f^{(N)}(x) dx, \quad j = N, \tag{3.4}$$

and

$$f_j = \frac{\int_0^1 f^{(j)}(x) dx + \sum_{k=0}^{N-j-1} \frac{2j!}{k+2} \binom{k+j+1}{k+1} E_{k+2}(0) f_{j+k+1}}{j!}, \tag{3.5}$$

$j = 0, 1, \dots, N - 2, N - 1.$

By replacing each term of (3.1) with it's approximation defined in (2.8), (2.11) and (3.3) respectively, and substituting  $x = x_k$  collocation points defined by

$$x_k = \frac{1}{N} k, \quad k = 0, 1, \dots, N,$$

we reach the following theorem.

**Theorem 3.1** *If the assumed approximate solution of the boundary-value problem (3.1), (1.2) is (2.5) then the discrete Euler system for the determination of the unknown coefficients  $\{c_n\}_{n=0}^N$  is given by*

$$\sum_{n=0}^N c_n E_n^{(2r)}(x_k) + \sum_{m=0}^{2r-1} \sum_{n=0}^N c_n P_m(x_k) E_n^{(m)}(x_k) = \sum_{n=0}^N f_n E_n(x_k). \tag{3.6}$$

*Proof* If we replace each term of (3.1) with its corresponding approximation given by (2.5), (3.2) and (3.3) and substitute with the collocation points  $x = x_k$  which is mentioned in section 2.2 and applying the collocation to it.  $\square$

Equation (3.6) can be written in matrix form

$$\mathbf{W} \mathbf{c} = \mathbf{F} \tag{3.7}$$

where

$$\mathbf{W} = \left[ \mathbf{M}^{2r} + \sum_{m=0}^{2r-1} \mathbf{P}_m \mathbf{M}^m \right]$$

and

$$\mathbf{P}_m = \begin{pmatrix} p_m(x_0) & 0 & 0 & \dots & 0 \\ 0 & p_m(x_1) & 0 & \dots & 0 \\ 0 & 0 & p_m(x_2) & 0 & 0 \\ \vdots & \vdots & \dots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & p_m(x_N) \end{pmatrix}.$$

The matrix representation of the boundary conditions (1.2), can be written as

$$\begin{aligned} \mathbf{M}^i \mathbf{E}(0) \mathbf{c} &= \alpha_i, \\ \mathbf{M}^i \mathbf{E}(1) \mathbf{c} &= \beta_i, \quad i = 0, 1, 2, \dots, r - 1. \end{aligned} \tag{3.8}$$

The simplification in conditions is done by writing (3.8) as

$$\Theta \mathbf{c} = \tilde{\mathbf{F}} \tag{3.9}$$

To obtain the solution of Eq. (3.1) with their conditions (1.2), we have to replace the row matrices (3.9) by the last  $2r$  rows of the matrix (3.7) and acquire the new augmented matrix

$$[\Theta, \tilde{\mathbf{F}}] = \begin{bmatrix} w_{00} & w_{01} & \dots & w_{0N} & \vdots & f_0 \\ w_{10} & w_{11} & \dots & w_{1N} & \vdots & f_1 \\ w_{20} & w_{21} & \dots & w_{2N} & \vdots & f_2 \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots \\ w_{N-2r,0} & w_{N-2r,1} & \dots & w_{N-2r,N} & \vdots & f_N - 2r \\ u_{00} & u_{01} & \dots & u_{0N} & \vdots & \alpha_0 \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots \\ u_{2r,0} & u_{2r,1} & \dots & u_{2r,N} & \vdots & \beta_{2r} \end{bmatrix}. \tag{3.10}$$

Now, we have a linear system of  $N + 1$  equations of the  $N + 1$  unknown coefficients. Calculations of those coefficients can be done by solving this linear system. There are numerous method for solving the system (3.10) among which is the Q-R method that will be used in this

paper. This solution produces the coefficients  $\mathbf{c}$  that is used for approximating Euler solution. The algorithm for the proposed method is listed below.

**Algorithm**

- Input number of iterations  $N$ ,
- Enter  $x = x_k$  collocation points,
- Calculate  $f_j$  from equations (3.4–3.5),
- Formulate the system  $\Theta \mathbf{c} = \tilde{\mathbf{F}}$ ,
- Solve the system using the Q-R method to find  $\mathbf{c}$ ,
- End.

**3.2 Nonlinear high-order BVP**

In this case we assign the source term  $\sigma(x, u) = f(x) - [u(x)]^n$  in Eq. (1.1) then by substituting each term in Eq. (3.1) with the approximation defined in (2.8),(2.11) and (3.3) respectively the equation becomes

$$\left[ \mathbf{M}^{2r} + \sum_{m=1}^{2r-1} P_m(x)\mathbf{M}^m \right] \mathbf{X}(x)\mathbf{B}^t \mathbf{c} + [\mathbf{X}(x) \mathbf{B}^t \mathbf{c}]^n = \mathbf{X}(x) \mathbf{B}^t \mathbf{F}. \tag{3.11}$$

where  $\mathbf{F} = [f_0, f_1, \dots, f_N]^T$  can be calculated also according to Eqs. (3.4)–(3.5). Then the residual term can be calculated

$$\mathfrak{R}_N(x) = \left[ \mathbf{M}^{2r} + \sum_{m=1}^{2r-1} P_m(x)\mathbf{M}^m \right] \mathbf{X}(x)\mathbf{B}^t \mathbf{c} + [\mathbf{X}(x) \mathbf{B}^t \mathbf{c}]^n - \mathbf{X}(x) \mathbf{B}^t \mathbf{F} \tag{3.12}$$

We need to first collocate Eq. (3.12) at  $N - 2r + 1$  points. For suitable collocation points we use the first  $N - 2r + 1$  Euler roots of  $E_{N+1}(x)$ . These equations obtained from Eq. (3.12) along with the boundary conditions from (1.2) generates  $N + 1$  nonlinear equations in  $N + 1$  unknowns coefficients  $\mathbf{c}$  that can be solved using Newton’s iterative method [11]. Consequently,  $u(x)$  can be calculated according to Eq. (2.5).

*Newton’s method*

In order to solve the system of Eq. (3.12), we formulate it as

$$\mathbf{R}(\mathbf{c}) = \begin{pmatrix} R_1(c_0, c_1, \dots, c_N) \\ R_2(c_0, c_1, \dots, c_N) \\ \vdots \\ R_{N+1}(c_0, c_1, \dots, c_N) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{3.13}$$

where  $\mathbf{c}$  is the column vector of the independent variables and  $\mathbf{R}$  is the column vector of the function  $\mathbf{R}_i$ , with  $\mathbf{R}_i(\mathbf{C}) = \mathbf{R}_i(c_0, c_1, \dots, c_N), 1 \leq i \leq N + 1$ . The number of zero valued functions is equal to the numbers of the independent variables. Then a good approximation to the method for solving Eq. (3.13) is the Newton’s method [11].

Consider  $\mathbf{c}^{(j)}$  be the initial guess at the iteration  $j$  of the solution. Also, let  $\mathbf{R}^{(j)}$  indicate the value of  $\mathbf{R}$  at the  $j$ th iteration. Assuming that  $\|\mathbf{R}^{(j)}\|$  is not too small, then we need to update vectors  $\Delta \mathbf{c}^{(j)}$

$$\mathbf{c}^{(j+1)} = \mathbf{c}^{(j)} + \Delta \mathbf{c}^{(j)} \iff \begin{pmatrix} c_0^{j+1} \\ c_1^{j+1} \\ \vdots \\ c_N^{j+1} \end{pmatrix} = \begin{pmatrix} c_0^j \\ c_1^j \\ \vdots \\ c_N^j \end{pmatrix} + \begin{pmatrix} \Delta c_0^j \\ \Delta c_1^j \\ \vdots \\ \Delta c_N^j \end{pmatrix} \tag{3.14}$$

such that  $\mathbf{R}(\mathbf{c}^{j+1}) = 0$ . Using the multidimensional extension of Taylor’s theorem for the approximation of the variation of  $\mathbf{R}(\mathbf{c})$  in the neighborhood of  $\mathbf{c}^j$  gives

$$\mathbf{R}(\mathbf{c}^j + \Delta \mathbf{c}^j) = \mathbf{R}(\mathbf{c}^{(j)}) + \mathbf{R}'(\mathbf{c}^{(j)})\Delta \mathbf{c}^j + o(\|\Delta \mathbf{c}^{(j)}\|^2) \tag{3.15}$$

where  $\mathbf{R}'(\mathbf{c}^{(j)})$  is the jacobian of the system of equations

$$\mathbf{R}'(\mathbf{c}) \equiv \mathbf{J}(\mathbf{c}) = \begin{pmatrix} \frac{\partial R_1}{\partial c_0}(\mathbf{c}) & \frac{\partial R_1}{\partial c_1}(\mathbf{c}) & \dots & \frac{\partial R_1}{\partial c_N}(\mathbf{c}) \\ \frac{\partial R_2}{\partial c_0}(\mathbf{c}) & \frac{\partial R_2}{\partial c_1}(\mathbf{c}) & \dots & \frac{\partial R_2}{\partial c_N}(\mathbf{c}) \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial R_{N+1}}{\partial c_0}(\mathbf{c}) & \frac{\partial R_{N+1}}{\partial c_1}(\mathbf{c}) & \dots & \frac{\partial R_{N+1}}{\partial c_N}(\mathbf{c}) \end{pmatrix}. \tag{3.16}$$

Appointing  $\mathbf{J}^{(j)}$  as the jacobian evaluated at  $\mathbf{c}^{(j)}$  and neglect terms of higher-order then we can rearrange Eq. (3.15) as

$$\mathbf{R}(\mathbf{c}^j + \Delta \mathbf{c}^j) = \mathbf{R}(\mathbf{c}^{(j)}) + \mathbf{J}^{(j)} \Delta \mathbf{c}^{(j)}. \tag{3.17}$$

Our goal of this method is to reach  $\mathbf{R}(\mathbf{c}^j + \Delta \mathbf{c}^j) = 0$ , so assigning that term to zero in the last equation gives

$$\mathbf{J}^{(j)} \Delta \mathbf{c}^{(j)} = -\mathbf{R}(\mathbf{c}^{(j)}). \tag{3.18}$$

Then, Eq. (3.18) is the system of  $N + 1$  linear equation in the  $N + 1$  unknown  $\Delta \mathbf{c}^{(j)}$ . Each Newton iteration step involves evaluating the vector  $\mathbf{R}^{(j)}$ , the matrix  $\mathbf{J}^{(j)}$  and a solution to Eq. (3.18). Newton iteration is stopped whenever the distance between two iteration is less than a give tolerance, i.e., when  $\|c^{(j+1)} - c^{(j)}\| \leq \epsilon$ . The algorithm for the above method is listed below.

**Algorithm**

- Initialize  $\mathbf{c} = \mathbf{c}^{(0)}$ ,
- For  $i = 0, 1, 2, \dots$ ,  $\mathbf{R}^{(j)}$  from equation (3.12),
- If  $\|\mathbf{R}^{(j)}\|$  is small enough, stop,
- Compute  $\mathbf{J}^{(j)}$ ,
- Solve  $\mathbf{J}^{(j)} \Delta \mathbf{c}^{(j)} = -\mathbf{R}(\mathbf{c}^{(j)})$ ,
- $\mathbf{c}^{(j+1)} = \mathbf{c}^{(j)} + \Delta \mathbf{c}^{(j)}$ ,
- End.

**4 Error analysis**

In this section, we will analyze the error of the presented method, suppose that  $H = L^2[0, 1]$ ,

$$\mathbf{U} = \text{span}\{E_0(x), E_1(x), \dots, E_N(x)\},$$



$f$  be an arbitrary element in  $H$  and  $\mathbf{U}$  is a finite dimensional vector space and  $f$  has the unique best approximation out of  $\mathbf{U}$  such that  $\bar{f} \in U$ , that is

$$\forall u \in \mathbf{U}, \quad \|f - \bar{f}\| \leq \|f - u\|.$$

Since  $\bar{f} \in \mathbf{U}$  is the best  $L_2$  approximation of  $f$  in  $\mathbf{U}$ , there exist a unique coefficients  $\{f_0, f_1, \dots, f_N\}$  such that

$$f(x) \simeq \bar{f}(x) = \sum_{n=0}^N f_n E_n(x) = \mathbf{E}(x)\mathbf{F},$$

where

$$\mathbf{E}(x) = [E_0(x), E_1(x), \dots, E_N(x)] \text{ and } \mathbf{F} = [f_0, f_1, \dots, f_N]^t.$$

**Theorem 4.1** [21] *Suppose that  $f(x) \in C^\infty[0, 1]$  and  $F(x)$  is the approximated function using Euler polynomials. Then the upper bound for these polynomials would be calculated according to the following*

$$\|E_N(x)\|_\infty \leq \mu N! \pi^{-(N+1)}$$

where  $\mu$  is a positive number independent of  $N$ .

**Theorem 4.2** *Suppose that  $u(x)$  be an enough smooth function and  $u_N(x)$  be the truncated Euler series of  $u(x)$ . Then,*

$$\|u(x) - u_N(x)\|_\infty \leq G N! (\pi)^{-(N+1)},$$

where  $G$  is a positive constant and the global error  $O(N! \pi^{-(N+1)})$ .

*Proof* In an operator form, Eq. (1.1) can be written as

$$Lu = u^{(2r)} = f(x) + g(x, u) \tag{4.1}$$

where the differential operator  $L$  is given by

$$L = \frac{d^{2r}}{dx^{2r}}.$$

The inverse operator  $L^{-1}$  is therefore considered a  $2r$ -fold integral operator defined by

$$L^{-1} = \underbrace{\int_0^x}_{(2r)\text{times}} (\cdot) \underbrace{dx}_{(2r)\text{times}}$$

Operating with  $L^{-1}$  on (4.1) yields

$$\begin{aligned} u(x) &= L^{-1} f(x) + L^{-1} g(x, u) \\ &= \underbrace{\int_0^x}_{(2r)\text{times}} f(x) \underbrace{dx}_{(2r)\text{times}} + \underbrace{\int_0^x}_{(2r)\text{times}} g(x, u) \underbrace{dx}_{(2r)\text{times}} \\ &= F(x) + G(x, u(x)). \end{aligned}$$

By approximating the functions  $u(x)$  and  $F(x)$  by the Euler polynomials. Therefore,

$$u_N(x) = F_N(x) + G(x, u_N(x)),$$

thus, by subtracting the last two equations we find

$$\begin{aligned} \|u(x) - u_N(x)\|_\infty &= \|F(x) - F_N(x) + (G(x, u(x)) - G(x, u_N(x)))\|_\infty \\ &= \|F(x) - F_N(x)\|_\infty + \|(G(x, u(x)) - G(x, u_N(x)))\|_\infty \quad (4.2) \\ \|u(x) - u_N(x)\|_\infty &\leq \|F(x) - F_N(x)\| + L_G \|u(x) - u_N(x)\|_\infty, \end{aligned}$$

where  $L_G$  is the Lipschitz constant of the function  $G(x, u(x))$ , then

$$\|u(x) - u_N(x)\|_\infty \leq \frac{1}{1 - L_G} \|F(x) - F_N(x)\|_\infty.$$

Using Theorem 4.1 yields

$$\|u(x) - u_N(x)\|_\infty \leq \frac{1}{1 - L_G} \left[ \mu N! \pi^{-(N+1)} \right] \leq \mathbf{G} N! \pi^{-(N+1)}.$$

where  $\mathbf{G} = \frac{\mu}{1 - L_G}$ . Thus the presented method supplies an approximate solution with global error  $O(N! \pi^{-(N+1)})$  which completes the proof. □

In the next section we will discuss the accuracy of the solution based on the residual function.

### 4.1 Accuracy of the solution based on residual function

We can easily check the accuracy of the suggested method as follows. Since the truncated Euler series in Eq. (2.6) is an approximate solution of Eq. (1.1), when the function  $u_N(x)$  and it's derivatives are substituted in Eq. (1.1), the resulting equation must be approximately satisfied ; that is, for

$$x = x_k \in [0, 1], \quad k = 0, 1, \dots, N.$$

$$\mathfrak{R}_N(x_k) = |u^{(2r)}(x) + \sum_{m=0}^{2r-1} P_m(x) u^{(m)}(x) - \sigma(x, u(x))| \cong 0. \quad (4.3)$$

or

$$\mathfrak{R}_N(x_k) \leq 10^{-qk}$$

where  $q_k$  is any positive integer. If  $\max 10^{-qk} = 10^{-q}$  ( $q$  is any positive integer) is prescribed, then the truncation limit  $N$  is increased until the difference  $\mathfrak{R}_N(x_k)$  at each of the points become smaller than the prescribed  $10^{-q}$ . Also, the error function can be estimated from

$$\mathfrak{R}_N(x) = u^{(2r)}(x) + \sum_{m=0}^{2r-1} P_m(x) u^{(m)}(x) - \sigma(x, u(x)).$$

If  $E_N(x) \rightarrow 0$ , as  $N$  has sufficiently enough value, then the error decreases.

### 5 Numerical examples

We present four numerical examples which will prove the accuracy, performance and effectiveness of the proposed techniques. We compare our results with the results that were collected from the open literature found in [12,23,24,28–30]. The computations associated with these examples were performed using Matlab 2014 on a personal computer. The absolute error may be defined as

$$\|e_N(x)\| = \max \|u(x) - u_N(x)\|, \quad 0 \leq x \leq 1 .$$

*Example 5.1* [30] Let us consider The linear 10th order boundary value problem according to Eq. (1.1) with  $r = 5$  in the form

$$u^{(10)} + 5u = 10 \cos(x) + 4(x - 1) \sin(x), \quad 0 \leq x \leq 1,$$

subject to the boundary conditions

$$\begin{aligned} u(0) &= 0, & u(1) &= 0, \\ u'(0) &= -1, & u'(1) &= \sin(1), \\ u''(0) &= 2, & u''(1) &= 2 \cos(1), \\ u'''(0) &= 1, & u'''(1) &= -3 \sin(1), \\ u^{(4)}(0) &= -4, & u^{(4)}(1) &= -4 \cos(1). \end{aligned}$$

whose exact solution is

$$u(x) = (x - 1) \sin x.$$

Maximum absolute error are tabulated in Table 1 at different values of  $N$  along with the CPU time. Table 2 exhibits a comparison between the errors obtained by using Euler matrix method with the analogous results of Viswanadham and Raju [30], for underlying B-spline method. Figure 2, illustrates the approximate and exact solution at  $N = 14$  and the error history at the same values.

*Example 5.2* [29,30] Now, we consider a nonlinear 10th-order boundary value problem

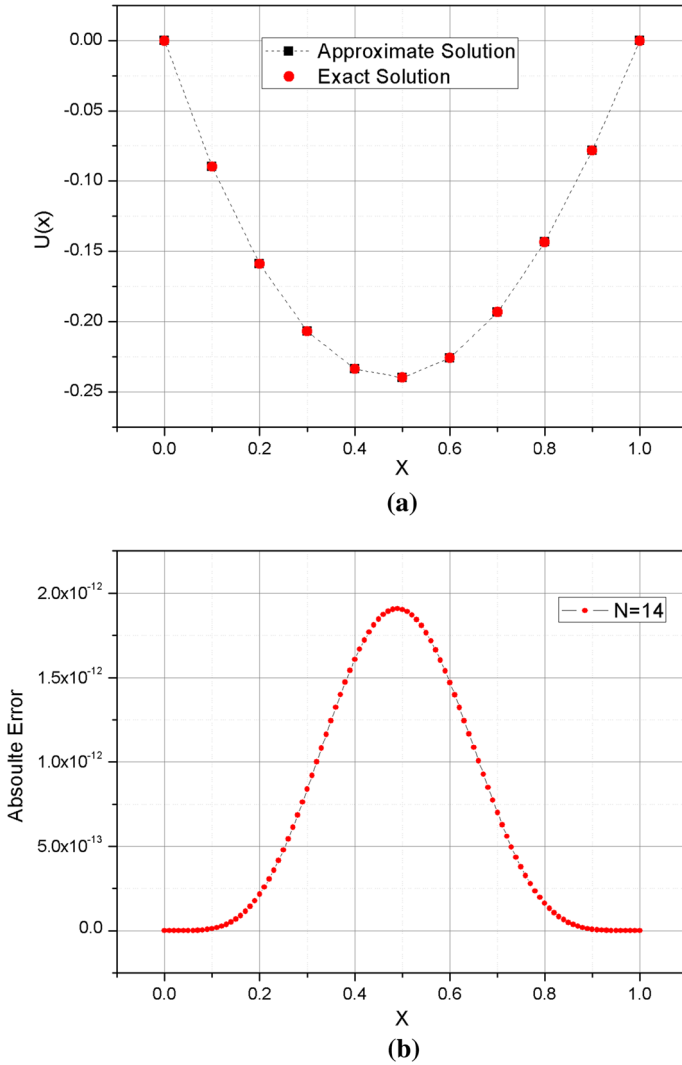
$$u^{(10)} = e^{-x} u^2$$

**Table 1** Maximum absolute errors and CPU time for Example 5.1

N	$\ e_N(x)\ $	CPU time (s)
10	3.05459E-10	2.36
12	3.30336E-11	2.65
14	1.90653E-12	2.75
16	1.56855E-12	3.24

**Table 2** Comparison of absolute error for Example 5.1

Presented method at N = 14	Quintic B-spline [30]
1.90653E-12	6.705523E-07



**Fig. 2** a Approximate and exact solution, b error history at  $N = 14$

subject to the boundary conditions

$$u(0) = u'(0) = u''(0) = u'''(0) = u^{(4)}(0) = 1,$$

$$u(1) = u'(1) = u''(1) = u'''(1) = u^{(4)}(1) = e,$$

whose exact solution is

$$u(x) = e^x.$$

Table 3 exhibits the maximum absolute error for different values of  $N$  along with the CPU time. Table 4 exhibits a comparison between the errors obtained by using Euler matrix method

**Table 3** Maximum absolute errors and CPU time for Example 5.2

N	$\ e_N(x)\ $	CPU time (s)
10	1.7891355E-10	3.007
12	3.9337866E-11	4.032
14	3.320233E-12	4.993
16	4.1944226E-13	6.056
18	5.7509557E-14	7.532

**Table 4** Comparison of absolute error for Example 5.2

Presented method at N=18	Iterative method [29]	Quintic B-spline [30]
5.7509557E-14	4.120E-09	2.14576E-06

**Table 5** Maximum absolute errors and CPU time for Example 5.3

N	$\ e_N(x)\ $	CPU time (s)
12	5.32157E-12	2.826
14	1.38123E-11	3.029
16	1.09609E-11	3.504

**Table 6** Comparison of absolute error for Example 5.3

Presented method at N = 14	1.38123E-11
Non-polynomial spline [28]	7.380E-09

with the analogous results of Rashidinia et al. [29], for underlying Iterative method and with the analogous results of Viswanadham, and Raju [30], for underlying B-spline method.

*Example 5.3* [28] Now we turn to linear 12th-order boundary value problem

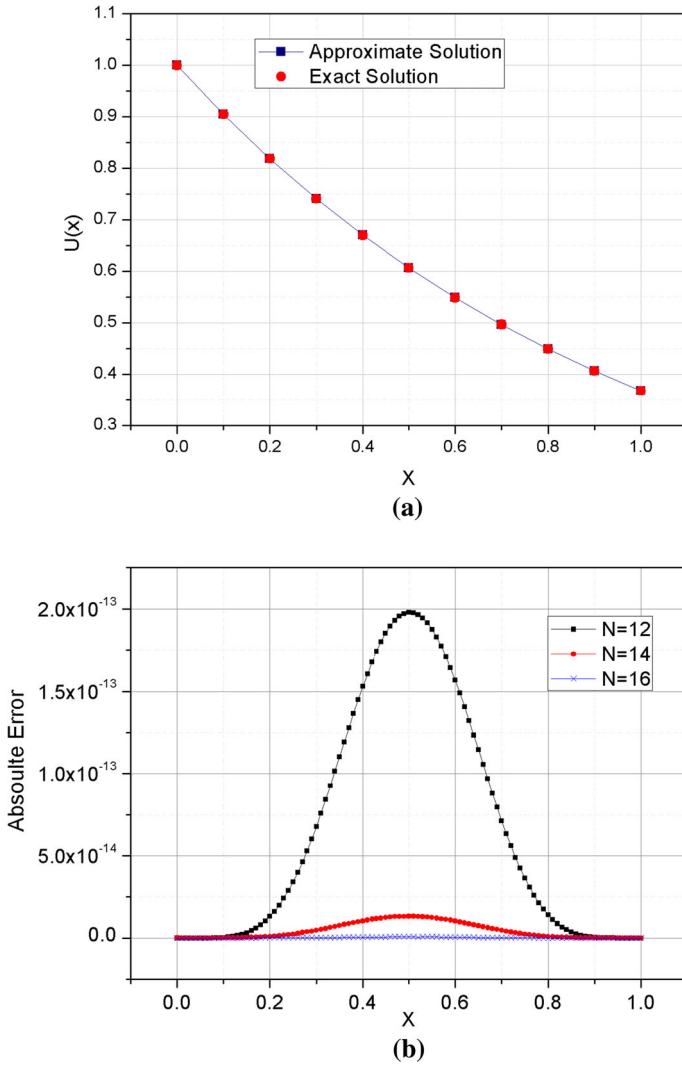
$$u^{(12)} + xu = -(120 + 23x + x^3)e^x, \quad 0 \leq x \leq 1,$$

**Table 7** Maximum absolute errors and CPU time for Example 5.4

N	$\ e_N(x)\ $	CPU time (s)
12	1.97953E-13	3.45
14	1.32117E-14	4.23
16	6.66134E-16	5.79
18	3.33067E-16	7.17

**Table 8** Comparison of maximum absolute error for Example 5.4

Presented method at N = 16	6.66134E-16
Iterative method [29]	3.220E-10
Differential transformation method [24]	1.610E-07
Homotopy perturbation method [12]	1.610E-07
Variational method [23]	1.612E-07



**Fig. 3** a Approximate and exact solution for Example 5.4, b comparison of error with  $N = 12, 14, 16$  for Example 5.4

subject to the boundary conditions

$$\begin{aligned}
 u(0) &= 0, & u(1) &= 0, \\
 u'(0) &= 1, & u'(1) &= -e, \\
 u''(0) &= 0, & u''(1) &= -4e, \\
 u'''(0) &= -3, & u'''(1) &= -9e, \\
 u^{(4)}(0) &= -8, & u^{(4)}(1) &= -16e, \\
 u^{(5)}(0) &= -15, & u^{(5)}(1) &= -25e,
 \end{aligned}$$

whose exact solution is

$$u(x) = x(1 - x)e^x.$$

Table 5 exhibits the maximum absolute error for different values of  $N$  along with the CPU time. Table 6 exhibits a comparison between the errors obtained by using Euler matrix method with the analogous results of Shahid [28], for underlying spline method.

*Example 5.4* [10, 12, 23, 24, 29] Our final example is a nonlinear 12th-order boundary value problem

$$u^{(12)} - u''' = 2e^x u^2, \quad 0 \leq x \leq 1,$$

subject to the boundary conditions

$$u^{(i)}(0) = (-1)^i, \quad u^{(i)}(1) = (-1)^i e^{-1}, \quad i = 0, 1, \dots, 5.$$

whose exact solution is

$$u(x) = e^{-x}.$$

Table 7 exhibits the maximum absolute error of the presented method at different values of  $N$  along with the CPU time. Also, Table 8 gives a comparison between the results obtained with the analogous results of Ullah et al. [29], for underlying Iterative method, with the analogous results of Opanuga et al. [24], for underlying differential transformation method, with the analogous results of Grover and Kumar [12] for underlying Homotopy perturbation method and with the analogous results of Noor and Mohyud-Din [23] for underlying variational method. Also, Fig. 3a introduces the maximum absolute error at different  $N$  and Fig. 3b exhibits the Euler solution and exact solutions.

## 6 Conclusion

This paper has discussed how Euler matrix collocation method can be applied for obtaining solutions of linear and nonlinear both tenth and twelfth-order boundary-value problems. The formulation and implementation of the scheme are illustrated. Our method is tested on four examples and comparisons with other methods are made. It is shown that Euler matrix method yields good results. Euler matrix method is a simple tool for providing high accurate numerical simulations to a large variety of model problems. So it is easily applied by researchers and engineers familiar with Euler matrix method.

**Acknowledgements** The authors are grateful for the referees for their valuable comments and suggestions on the original manuscript.

## References

1. Abramowitz, M., Stegun, I.: Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, National Bureau of Standards. Wiley, New York (1972)
2. Agarwal, R.: Boundary value problems for high ordinary differential equations. World Scientific, Singapore (1986)
3. Alhayani, W.: Adomian decomposition method with Greens function for solving twelfth-order boundary value problems. Appl. Math. Sice. **9**, 353–368 (2015)

4. Bildik, N., Tosun, M., Deniz, S.: Euler matrix method for solving complex differential equations with variable coefficients in rectangular domains. *Int. J. Appl. Phys. Math.* **7**, 69–78 (2017)
5. Bishop, R., Cannon, S., Miao, S.: On coupled bending and torsional vibration of uniform beams. *J. Sound Vib.* **131**, 457–464 (1989)
6. Boutayeb, A., Twizell, E.: Finite-difference methods for the solution of special eighth-order boundary value problems. *Int. J. Comput. Math.* **48**, 63–75 (1993)
7. Brettii, G., Ricci, P.: Euler polynomials and the related quadrature rule. *Ga. Math. J.* **8**, 447–453 (2001)
8. Chandrasekhar, S.: *Hydrodynamic and Hydromagnetics Stability*. Dover, New York (1981)
9. Djidjeli, K., Twizell, E., Boutayeb, A.: Numerical methods for special nonlinear boundary value problems of order  $2m$ . *J. Comput. Math. Appl.* **47**, 35–45 (1993)
10. El-Gamel, M.: Chebyshev polynomial solutions of twelfth-order boundary-value problems. *Br. J. Math. Comput. Sci.* **6**, 13–23 (2015)
11. El-Gamel, M., Zayed, A.: Sinc-Galerkin method for solving non-linear boundary value problems. *Comput. Math. Appl.* **48**, 1285–1298 (2004)
12. Grover, M., Kumar, A.: A New approach to evaluate 12th order boundary value problems with HPM. *Int. J. Comput. Math. Numer. Simul.* **4**, 99–112 (2011)
13. Hansen, E.: *A Table of Series and Products*. Eaglewood cliffs, Prentice hall (1975)
14. Lehmer, D.: A new approach to Bernoulli polynomials. *Am. Math. Mon.* **95**, 905–911 (1988)
15. Magnus, W., Oberhettinger, F., Soni, R.: *Formulas and Theorems for the Special Functions of Mathematical Physics*. Springer, New York (1966)
16. Mirzaee, F., Bimesl, S.: An efficient numerical approach for solving systems of high-order linear Volterra integral equations. *Sci. Iran. D* **21**, 2250–2263 (2014)
17. Mirzaee, F., Bimesl, S.: Solving systems of high-order linear differential difference equations via Euler matrix method. *J. Egypt. Math. Soc.* **23**, 286–291 (2015)
18. Mirzaee, F., Bimesl, S.: Numerical solutions of systems of high-order Fredholm integro-differential equations using Euler polynomials. *Appl. Math. Model.* **39**, 6767–6769 (2015)
19. Mirzaee, F., Bimesl, S., Tohidi, E.: A numerical framework for solving high-order pantograph delay Volterra integro-differential equations. *Kuwait J. Sci.* **43**, 69–83 (2016)
20. Mirzaee, F., Hoseini, S.: A new collocation approach for solving systems of high-order linear Volterra integro-differential equations with variable coefficients. *Appl. Math. Comput.* **311**, 272–282 (2017)
21. Mirzaee, F., Bimesl, S.: A uniformly convergent Euler matrix method for telegraph equations having constant coefficients. *Mediterr. J. Math.* **13**, 469–515 (2014)
22. Natalin, P., Bernaridini, A.: A generalization of the Bernoulli polynomials. *J. Appl. Math.* **3**, 155–163 (2003)
23. Noor, M., Mohyud-Din, S.: Solution of twelfth-order boundary value problems by variational iteration technique. *J. Appl. Math. Comput.* **28**, 123–131 (2008)
24. Opanuga, A., Okagbue, H., Edeki, S., Agboola, O.: Differential transform technique for higher order boundary value problems. *Mod. Appl. Sci.* **9**, 224–230 (2015)
25. Pandey, R., Kumar, N., Bhardwaj, A., Dutta, G.: Solution of Lane-Emden type equations using Legendre operational matrix of differentiation. *App. Math. Comput.* **218**, 7629–7637 (2012)
26. Rashidinia, J., Jalilian, R., Farajeyan, K.: Non polynomial spline solutions for special linear tenth-order boundary value problems. *World J. Model. Simul.* **7**, 40–51 (2011)
27. Roman, S.: *The Umbral Calculus*. Academic Press, New York (1984)
28. Shahid, S., Siddiqi, S., Akram, G.: Solutions of 12th order boundary value problems using non-polynomial spline technique. *Appl. Math. Comput.* **199**, 559–571 (2008)
29. Ullah, I., Khan, H., Rahim, M.: Numerical solutions of higher order nonlinear boundary value problems by new iterative method. *Appl. Math. Sci.* **7**, 2429–2439 (2013)
30. Viswanadham, K., Raju, Y.: Quintic B-spline collocation method for tenth order boundary value problems. *Int. J. Comput. Appl.* **51**, 975–989 (2012)
31. Wazwaz, A.: The modified Adomian decomposition method for solving linear and nonlinear boundary value problems of tenth-order and twelfth-order. *Int. J. Nonlinear Sci. Numer. Simul.* **1**, 17–24 (2000)
32. Young, P.: Congruences for Bernoulli, Euler and Stirling numbers. *Int. J. Number Theory* **78**, 204–227 (1999)