

RESEARCH

Open Access



# The refinement-schemes-based unified algorithms for certain $n$ th order linear and nonlinear differential equations with a set of constraints

Syeda Tehmina Ejaz<sup>1</sup>, Ghulam Mustafa<sup>2</sup> , Dumitru Baleanu<sup>3,4</sup> and Yu-Ming Chu<sup>5,6\*</sup>

\*Correspondence:

[chuyuming2005@126.com](mailto:chuyuming2005@126.com)

<sup>5</sup>Department of Mathematics, Huzhou University, Huzhou, 313000, P.R. China

<sup>6</sup>Hunan Provincial Key Laboratory of Mathematical Modeling and Analysis in Engineering, Changsha University of Science and Technology, Changsha, 410114, P.R. China

Full list of author information is available at the end of the article

## Abstract

We first present a generalized class of binary interpolating refinement schemes and their properties. Then the refinement-schemes-based unified algorithms for the solution of certain  $n$ th order linear and nonlinear differential equations with a set of constraints are presented. Moreover, several algorithms based on the refinement schemes for solving differential equations are the special cases of our algorithms.

**MSC:** 65L10; 65L99; 65D07; 65M70; 65G50

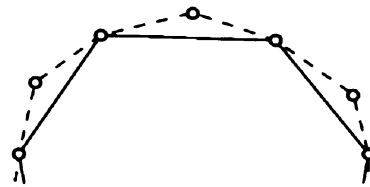
**Keywords:** Differential equation; Constraints; Boundary value problems; Algorithm; Refinement schemes; Interpolation

## 1 Introduction

The refinement schemes, also known as subdivision schemes, are efficient tools for the modeling of curves. These schemes are classified into two main categories, interpolating and approximating. These categories are further classified into  $n$  subcategories: binary, ternary, ...,  $n$ -ary. In this paper, we focus on binary interpolating refinement schemes. The domain of these schemes is a polygon while the range is a refined polygon. These schemes have two main rules, namely refinement and topological rules. There are two refinement rules: one rule carries on the points of a coarse polygon while the other rule introduces the new points corresponding to each edge of the polygon. These rules are called even and odd rules, respectively. The even rule just carries on the old points. The odd rule is an affine combination of the points of the coarse polygon. Furthermore, the topological rule is just the connection of adjacent new and old points with straight lines. The topological and refinement rules give us a new polygon. The repeated application of the refinement and topological rules gives a smooth shape. Graphically, this procedure is depicted in Fig. 1. Mathematically, if  $Q^k = \{Q_i^k\}_{i \in \mathbb{Z}}$  is a polygon at the  $k$ th level then the refined polygon  $Q^{k+1} =$

© The Author(s) 2021. This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

**Figure 1** Refinement procedure. The solid lines show the coarse polygon while the dotted lines show the refined polygon



$\{Q_i^{k+1}\}_{i \in \mathbb{Z}}$  can be obtained by applying the topological and refinement rules [1] as follows:

$$\begin{cases} \text{(Even rule)} & Q_{2i}^{k+1} = Q_i^k, \\ \text{(Odd rule)} & Q_{2i+1}^{k+1} = \sum_{j=0}^{n_1} \chi_{n_1,j} (Q_{i-j}^k + Q_{i+j+1}^k), \end{cases} \quad (1)$$

where the coefficients appearing in the affine combination of points are

$$\chi_{n_1,j} = \frac{((2n_1 + 1)!!)^2}{2(4^{n_1})(2n_1 + 1)!} \frac{(-1)^j}{(2j + 1)} \binom{2n_1 + 1}{n_1 - j}, \quad j = 0, 1, 2, \dots, n_1, \quad (2)$$

while  $\binom{2n_1+1}{n_1-j}$  denotes the binomial coefficient. Here  $n_1$  is the complexity of the scheme. If  $n_1 = 0, 1, 2, 3, \dots$ , then the complexity of the scheme will be 2, 4, 6, 8,  $\dots$ , respectively. In other words, for  $n_1 = 0, 1, 2, 3$ , we get 2-, 4-, 6-, and 8-point schemes, respectively.

The refinement procedure has attracted attention due to a large variety of applications in curve modeling and algorithms for the solution of differential equations with a set of constraints. Mathematically, these equations are called the boundary value problems (BVPs). Higher-order linear and nonlinear differential equations have been reported in mathematical physics and structural engineering. Different techniques have been introduced to solve such problems. Here is a list of works that have caught the attention of the scientific community, pointing to the diversity of the applications of refinement schemes in the area of differential equations.

In 1996, initially, Qu and Agarwal [2] presented a refinement-scheme-based algorithm for the second-order linear differential equations (DEs). A year later, Qu and Agarwal [3] offered an algorithm for the second-order nonlinear DEs. Then after the long silence, Mustafa and Ejaz [4] introduced the refinement-based algorithm for the third-order linear DEs in 2014. In 2015, Ejaz et al. [5] introduced an algorithm for the fourth-order linear DEs. Ejaz and Mustafa [6] offered an algorithm for the third-order nonlinear DEs in 2016. In the next year, Mustafa et al. [7] introduced an algorithm for the fourth-order nonlinear DEs.

In this paper, we present generalized algorithms based on generalized refinement schemes for the  $n$ th order linear and nonlinear DEs. We prove that all the above algorithms are special cases of our generalized algorithms. We consider the following two-point  $n$ th order linear and nonlinear DEs with a set of constraints

$$\text{(Linear DE)} \quad u^n(t) + q(t)u(t) = f(t), \quad a \leq t \leq b, \quad (3)$$

and

$$\text{(Nonlinear DE)} \quad u^n(t) = f(t, u(t), u'(t)), \quad (4)$$

where the set of constraints is defined as follows: If  $n$  is even, a set of constraints is

$$u^l(a) = \alpha_l, \quad u^m(b) = \beta_m, \quad l, m = 0, 1, 2, \dots, n - 2; \tag{5}$$

If  $n$  is odd, a set of constraints is

$$u^l(a) = \alpha_l, \quad u^{m-1}(b) = \beta_m, \quad l, m = 0, 1, 2, \dots, n - 2,$$

or (6)

$$u^{l-1}(a) = \alpha_l, \quad u^m(b) = \beta_m, \quad l, m = 0, 1, 2, \dots, n - 2,$$

where  $\alpha_l$  and  $\beta_m$  are scalars. We assume that the problems are well-posed throughout the paper.

The rest of the work is structured as follows. In Sect. 2, we discuss the properties of generalized binary interpolating refinement schemes. We also present generalized formulae for the  $n$ th derivatives of the refinement schemes in this section. The generalized algorithms for the  $n$ th order linear and nonlinear DEs are presented in Sect. 3. In Sect. 4, we present the generalized form of imposed constraints and the approximation of derivative involved in the constraints. In Sect. 5, we discuss the stable linear and nonlinear system of equations. We also discuss the existence of the solutions of these systems in this section. In Sect. 6, we show that the refinement-based existing algorithms are special cases of our generalized algorithms. Section 7 presents the conclusion.

### 2 Properties of the refinement scheme

The  $n$ th order continuous (i.e.,  $C^n$  continuous) refinement scheme is suitable to develop an algorithm for the solution of the  $n$ th order DEs. For example, if we want to find solutions of the eighth order DEs with a set of constraints then we have to choose a  $C^8$ -continuous refinement scheme from (1).

Here we briefly summarize the continuity and other properties of a refinement scheme. If  $\{Q_i = (i, \delta_0)^T\}$  is the initial data then repeated application of the scheme produces the limit curve named  $\rho(t)$ , also known as a basis function, where

$$\rho(i) = \begin{cases} 1, & i = 0, \\ 0, & i \neq 0. \end{cases} \tag{7}$$

and

$$\rho(t) = \rho_{n_1}(t) = \rho(2t) + \sum_{j=-n_1}^{n_1} \chi_{n_1, |j|} \rho(2t - 2j + 1), \quad t \in \mathbb{R}. \tag{8}$$

The scheme (1) has the following properties:

- It produces  $C^n$  continuous curves, where  $n = n_1$  for  $n_1 \leq 5$ , and for a large value of  $n_1$ ,  $n = 0.415n_1$  by [8]. It means that  $\rho(t)$  is  $n$  times continuously differentiable.
- Its degree of generation and reproduction is  $2n_1 + 1$ .
- The approximation order of the scheme is  $2n_1 + 2$ .
- The support of  $\rho(t)$  is finite; explicitly, its support is  $(-2n_1 - 1, 2n_1 + 1)$ .

- The iteration matrix, also known as a local refinement matrix, is defined in (9).

$$S = \begin{pmatrix}
 0 & 0 & 0 & \dots & 0 & 1 & 0 & 0 & 0 & \dots \\
 \chi_{n_1, n_1} & \chi_{n_1, n_1-1} & \chi_{n_1, n_1-2} & \dots & \chi_{n_1, -1} & \chi_{n_1, 0} & \chi_{n_1, 0} & \chi_{n_1, -1} & \chi_{n_1, -2} & \dots \\
 0 & 0 & 0 & \dots & 0 & 0 & 1 & 0 & 0 & \dots \\
 0 & \chi_{n_1, n_1} & \chi_{n_1, n_1-1} & \dots & \chi_{n_1, -2} & \chi_{n_1, -1} & \chi_{n_1, 0} & \chi_{n_1, 0} & \chi_{n_1, -1} & \dots \\
 0 & 0 & 0 & \dots & 0 & 0 & 0 & 1 & 0 & \dots \\
 0 & 0 & \chi_{n_1, n_1} & \dots & \chi_{n_1, -3} & \chi_{n_1, -2} & \chi_{n_1, -1} & \chi_{n_1, 0} & \chi_{n_1, 0} & \dots \\
 0 & 0 & 0 & \dots & 0 & 0 & 0 & 1 & 0 & \dots \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots \\
 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots \\
 \\
 0 & 0 & 0 & 0 & \dots & 0 & 0 & \\
 \chi_{n_1, n_1-2} & \chi_{n_1, n_1-1} & \chi_{n_1, n_1} & 0 & \dots & 0 & 0 & \\
 0 & 0 & 0 & 0 & \dots & 0 & 0 & \\
 \chi_{n_1, n_1-3} & \chi_{n_1, n_1-2} & \chi_{n_1, n_1-1} & \chi_{n_1, n_1} & \dots & 0 & 0 & \\
 \dots & 0 & 0 & 0 & \dots & 0 & 0 & \\
 \chi_{n_1, n_1-4} & \chi_{n_1, n_1-3} & \chi_{n_1, n_1-2} & \chi_{n_1, n_1-1} & \dots & 0 & 0 & \\
 \dots & 0 & 0 & 0 & \dots & 0 & 0 & \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\
 0 & 0 & 0 & 0 & \dots & \chi_{n_1, n_1-1} & \chi_{n_1, n_1} & \\
 0 & 0 & 0 & 0 & \dots & 0 & 0 &
 \end{pmatrix}. \tag{9}$$

The order of the refinement matrix is  $4n_1 + 1$ .

- The matrix  $S$  has the following  $k$  eigenvalues  $\lambda_k$  and their corresponding right-eigenvectors  $R_k$ :

$$\begin{aligned}
 \lambda_k &= 2^{-k}, \quad 0 \leq k \leq 2n_1 + 1, \\
 R_k &= ((-2n_1)^k, (-2n_1 + 1)^k, \dots, (-2)^k, (-1)^k, 0, 1, 2^k, \\
 &\quad \dots, (2n_1 - 1)^k, (2n_1)^k)_{(4n_1+1) \times 1}^T.
 \end{aligned} \tag{10}$$

- The left-eigenvectors  $L_k$  corresponding to the eigenvalues  $\lambda_k$  can be found by using the transpose of local refinement matrix  $S$ . These eigenvectors also satisfy the relation  $R_i^T L_j = \delta_i \delta_j, \forall i, j: i, j = 1, 2, \dots, n_1$ . That is,

$$\begin{aligned}
 R_i^T L_j &= 1, \quad \text{if } i = j, \\
 R_i^T L_j &= 0, \quad \text{if } i \neq j.
 \end{aligned}$$

**Proposition 1** *The function  $\rho(t)$  is  $n$ -times continuously differentiable on the interval  $(-2n_1 - 1, 2n_1 + 1)$  and its  $n$ th derivatives are given by*

$$\rho^n(t) = 2^n (\text{sgn}(t))^n E_{|t|}^T L_n, \quad -2n_1 \leq t \leq 2n_1,$$

where

$$\operatorname{sgn}(t) = \begin{cases} -1 & \text{if } t < 0, \\ 0 & \text{if } t = 0, \\ 1 & \text{if } t > 0, \end{cases}$$

$$E_t = (a_{(2n)t}, a_{(2n-1)t}, \dots, a_{1t}, a_{0t}, a_{-1t}, a_{-2t}, \dots, a_{(-2n+2)t}, a_{(-2n+1)t}, a_{(-2n)t}),$$

for  $0 \leq t \leq 4n_1 + 1$  and

$$a_{pt} = \begin{cases} 1 & \text{if } p = t, \\ 0 & \text{if } p \neq t. \end{cases}$$

### 3 Generalized algorithms for the $n$ th order linear and nonlinear DEs

We structure the refinement-schemes-based algorithms for the two-point  $n$ th order linear and nonlinear DEs with the set of constraints (3) and (4).

#### 3.1 Generalized algorithm for the $n$ th order linear DEs

In this section, we construct the refinement-scheme-based algorithm for the  $n$ th order linear DEs. Let the solution of (3) be

$$G_L(t) = \sum_{i=-2n_1}^{N+2n_1} g_i^L \rho\left(\frac{t-t_i}{h}\right), \tag{11}$$

where  $a \leq t \leq b$ ,  $N \geq 2n_1$ ,  $t_i = ih$ ,  $g_i^L = G_L(t_i)$ , and  $h = \frac{b-a}{N}$ .

This implies by (3) that

$$G_L^n(t_j) + q(t_j)G_L(t_j) = f(t_j), \quad j = 0, 1, \dots, N. \tag{12}$$

If we have a set of constraints as in (5) then we get

$$G_L^l(a) = \alpha_l, \quad G_L^m(b) = \beta_m, \quad l, m = 0, 1, 2, \dots, n-2. \tag{13}$$

For a set of constraints as in (6), it takes the form

$$G_L^l(a) = \alpha_l, \quad G_L^{m-1}(b) = \beta_m, \quad l, m = 0, 1, 2, \dots, n-2, \tag{14}$$

or

$$G_L^{l-1}(a) = \alpha_l, \quad G_L^m(b) = \beta_m, \quad l, m = 0, 1, 2, \dots, n-2.$$

From (11), we have

$$G_L'(t_j) = \frac{1}{h} \sum_{i=-2n_1}^{N+2n_1} g_i^L \rho'\left(\frac{t_j-t_i}{h}\right),$$

$$G_L''(t_j) = \frac{1}{h^2} \sum_{i=-2n_1}^{N+2n_1} g_i^L \rho''\left(\frac{t_j-t_i}{h}\right),$$

$$\vdots$$

$$G_L^n(t_j) = \frac{1}{h^n} \sum_{i=-2n_1}^{N+2n_1} g_i^L \rho^n \left( \frac{t_j - t_i}{h} \right). \tag{15}$$

Now using (15) in (12), we get

$$\sum_{i=-2n_1}^{N+2n_1} g_i^L \rho^n \left( \frac{t_j - t_i}{h} \right) + h^n q(t_j) \sum_{i=-2n_1}^{N+2n_1} g_i^L \rho \left( \frac{t_j - t_i}{h} \right) = h^n f(t_j), \quad j = 0, 1, \dots, N.$$

This implies

$$\sum_{i=-2n_1}^{N+2n_1} g_i^L \rho^n(j - i) + h^n q(t_j) \sum_{i=-2n_1}^{N+2n_1} g_i^L \rho(j - i) = h^n f(t_j), \quad j = 0, 1, \dots, N.$$

Let  $\rho_i^n = \rho^n(i)$ , then

$$\sum_{i=-2n_1}^{N+2n_1} g_i^L \rho_{j-i}^n + h^n q(t_j) \sum_{i=-2n_1}^{N+2n_1} g_i^L \rho_{j-i} = h^n f(t_j), \quad j = 0, 1, \dots, N.$$

This can be simplified as

$$\sum_{i=-2n_1}^{N+2n_1} g_i^L [\rho_{j-i}^n + h^n q(t_j) \rho_{j-i}] = h^n f(t_j), \quad j = 0, 1, \dots, N. \tag{16}$$

This system has  $N + 1$  equations. Its matrix is as follows:

$$A_L G_L = d_L, \tag{17}$$

where the banded matrix  $(A_L)_{(N+1) \times (N+4n_1+1)}$ , column vectors  $(G_L)_{(N+4n_1+1)}$  and  $(d_L)_{(N+1)}$  are defined in (18), (19), and (20), respectively,

$$A_L = \begin{pmatrix} \rho_{-2n_1}^n & \rho_{-2n_1+1}^n & \rho_{-2n_1+2}^n & \cdots & \rho_{-1}^n & \chi_0 & \rho_1^n & \rho_2^n & \cdots & \rho_{2n_1}^n & 0 \\ 0 & \rho_{-2n_1}^n & \rho_{-2n_1+1}^n & \cdots & \rho_{-2}^n & \rho_{-1}^n & \chi_1 & \rho_1^n & \cdots & \rho_{2n_1-1}^n & \rho_{2n_1}^n \\ 0 & 0 & \rho_{-2n_1}^n & \cdots & \rho_{-3}^n & \rho_{-2}^n & \rho_{-1}^n & \chi_2 & \cdots & \rho_{2n_1-2}^n & \rho_{2n_1-1}^n \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \\ 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \rho_{2n_1}^n & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & \dots & \dots & \rho_{2n_1-1}^n & \rho_{2n_1-1}^n \end{pmatrix}, \tag{18}$$

and  $\chi_j = \rho_j^n + q_j h^n$ ,

$$G_L = (g_{-2n_1}^L, g_{-2n_1+1}^L, \dots, g_{2n_1+N-1}^L, g_{2n_1+N}^L)^T, \tag{19}$$

and

$$d_L = (h^n f_0, h^n f_1, \dots, h^n f_N)^T. \tag{20}$$

The system (17) is unstable. To find the solution of (17), we need  $4n_1$  more equations. As the  $n$  constraints are given in (3), so we only need to construct  $4n_1 - n$  constraints. The remaining  $4n_1 - n$  constraints can be found by some extrapolation method.

For this, we have the following two cases:

- If  $4n_1 - n$  is even, then we construct  $\frac{4n_1-n}{2}$  constraints at the left end of the DE and  $\frac{4n_1-n}{2}$  constraints on the right end.
- If  $4n_1 - n$  is odd then we construct  $\frac{4n_1-n+1}{2}$  constraints at the left end and  $\frac{4n_1-n-1}{2}$  constraints at the right end of the DE.

The treatment of these constraints is given in the next section.

### 3.2 Generalized algorithm for the $n$ th order nonlinear DEs

In this subsection, we construct the refinement-scheme-based algorithm for the  $n$ th order nonlinear DEs. Let the solution of (4) be

$$G_{NL}(t) = \sum_{i=-2n_1}^{N+2n_1} g_i^{NL} \rho \left( \frac{t-t_i}{h} \right), \tag{21}$$

where  $a \leq t \leq b, N \geq 2n_1, t_i = ih, g_i^{NL} = G_{NL}(t_i),$  and  $h = \frac{b-a}{h}$ .

This implies by (4) that

$$G_{NL}^n(t_j) = f(t_j, G_{NL}(t_j), G'_{NL}(t_j), \dots, G_{NL}^{n-1}(t_j)), \quad j = 0, 1, \dots, N, \tag{22}$$

with the set of constraints given in (13) or (14). From (21), we have

$$\begin{aligned} G'_{NL}(t_j) &= \frac{1}{h} \sum_{i=-2n_1}^{N+2n_1} g_i^{NL} \rho' \left( \frac{t_j-t_i}{h} \right), \\ G''_{NL}(t_j) &= \frac{1}{h^2} \sum_{i=-2n_1}^{N+2n_1} g_i^{NL} \rho'' \left( \frac{t_j-t_i}{h} \right), \\ &\vdots \\ G_{NL}^n(t_j) &= \frac{1}{h^n} \sum_{i=-2n_1}^{N+2n_1} g_i^{NL} \rho^n \left( \frac{t_j-t_i}{h} \right). \end{aligned} \tag{23}$$

Now using (23) in (22), we get

$$\sum_{i=-2n_1}^{N+2n_1} g_i^{NL} \rho^n \left( \frac{t_j-t_i}{h} \right) = h^n f(t_j, G_{NL}(t_j), G'_{NL}(t_j), \dots, G_{NL}^{n-1}(t_j)), \quad j = 0, 1, \dots, N.$$

This implies

$$\sum_{i=-2n_1}^{N+2n_1} g_i^{NL} \rho^n(j-i) = h^n f(t_j, G_{NL}(t_j), G'_{NL}(t_j), \dots, G_{NL}^{n-1}(t_j)), \quad j = 0, 1, \dots, N.$$

This leads to

$$\sum_{i=-2n_1}^{N+2n_1} g_i \rho_{j-i}^n = h^n f(t_j, G_{NL,j}, G'_{NL,j}, \dots, G_{NL,j}^{n-1}), \quad j = 0, 1, \dots, N, \tag{24}$$

where  $G_{NL,j}^l = G_{NL}^l(t_j)$ , for  $l = 0, 1, \dots, n - 1$ . This system also has  $N + 1$  equations. Its matrix form is

$$A_{NL} G_{NL} = d_{NL}, \tag{25}$$

where  $A_{NL}$  is the banded matrix of order  $(N + 1) \times (N + 4n_1 + 1)$ ,  $G_{NL}$  and  $d_{NL}$  have orders  $(N + 4n_1 + 1) \times 1$  and  $(N + 1) \times 1$ , respectively. These matrices are defined below.

$$A_{NL} = \begin{pmatrix} \rho_{-2n_1}^n & \rho_{-2n_1+1}^n & \rho_{-2n_1+2}^n & \dots & \rho_{-1}^n & \rho_0 & \rho_1^n & \rho_2^n & \dots & \rho_{2n_1}^n & 0 \\ 0 & \rho_{-2n_1}^n & \rho_{-2n_1+1}^n & \dots & \rho_{-2}^n & \rho_{-1}^n & \rho_1 & \rho_1^n & \dots & \rho_{2n_1-1}^n & \rho_{2n_1}^n \\ 0 & 0 & \rho_{-2n_1}^n & \dots & \rho_{-3}^n & \rho_{-2}^n & \rho_{-1}^n & \rho_2 & \dots & \rho_{2n_1-2}^n & \rho_{2n_1-1}^n \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & \dots \\ 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \rho_{2n_1}^n & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & \dots & \dots & \rho_{2n_1-1}^n & \rho_{2n_1-1}^n \end{pmatrix}, \tag{26}$$

This can be simplified as

$$A_{NL} = (-1)^n (\rho_{rc}^n (c - r - 2n_1 - 1)),$$

where  $r = 1, 2, \dots, N$  and  $c = 1, 2, \dots, N + 4n_1 + 1$ . The column matrices are defined as

$$G_{NL} = (g_{-2n_1}, g_{-2n_1+1}, \dots, g_{2n_1+N-1}, g_{2n_1+N})^T \tag{27}$$

and

$$d_{NL} = h^n (f_0, f_1, \dots, f_{N-1}, f_N)^T, \tag{28}$$

where  $f_j = f(t_j, g_{NL,j}, g'_{NL,j}, \dots, g_{NL,j}^{n-1})$  and  $g_{NL,j}^l = g_{NL}^l(t_j)$ , for  $j = 0, 1, \dots, n$  and  $l = 0, 1, \dots, n - 1$ . To find the solution of (25), we need  $4n_1$  more equations to solve the system (25). As the  $n$  constraints are given in equation (4), so we only need to construct  $4n_1 - n$  constraints. The remaining  $4n_1 - n$  constraints can be found by some extrapolation method, the detail is given below.

- If  $4n_1 - n$  is even, then we construct  $\frac{4n_1-n}{2}$  constraints at the left end of the DE and  $\frac{4n_1-n}{2}$  constraints on the right end.
- If  $4n_1 - n$  is odd then we construct  $\frac{4n_1-n+1}{2}$  constraints at the left end and  $\frac{4n_1-n-1}{2}$  constraints at the right end of the DE.



#### 4 Approximation of the given and imposed constraints

The given derivative constraints and the imposed constraints for the unstable systems (17) and (25) are approximated in this section.

##### 4.1 Approximation of the given derivative constraints

If  $F(t)$  is a function then for  $h > 0$  and integer  $p > 0$ , the  $l$ th derivative of  $G(t)$  can be approximated by the finite difference method as

$$G^l(t) = \frac{l!}{h^l} \sum_{i=i_{\min}}^{i_{\max}} c_i F(t + ih) + O(h^p). \tag{29}$$

The necessary condition for (29) to be satisfied is

$$\sum_{i=i_{\min}}^{i_{\max}} i^n c_i = \begin{cases} 0, & \text{for } 0 \leq n \leq l + p - 1 \text{ and } n \neq l, \\ 1, & \text{for } n = l. \end{cases} \tag{30}$$

For  $i_{\min} = 0$  and  $i_{\max} = l + p - 1$ , the forward difference approximation can be expected. The convolution mask is the vector  $C = (c_{i_{\min}}, \dots, c_{i_{\max}})$ . If we solve the system (30) then we get the convolution matrix  $C$ .

##### 4.2 Approximation of the imposed constraints

We impose the set of constraints on the left as well as on the right side of the given constraints of the linear and nonlinear DE. These constraints are constructed as follows.

###### 4.2.1 The left end imposed constraints

If  $S_1(t)$  is the polynomial which interpolates the data  $(t_i, g_i), 0 \leq i \leq 2n_1 - 1$ , then the values  $g_{-(2n_1-1)}, g_{-(2n_1)}, \dots, g_{-2}, g_{-1}$  imposed on the left side can be found. That is,

$$g_{-i} = S_1(-t_i), \quad i = \begin{cases} 1, 2, \dots, \frac{4n_1-n}{2}, & \text{if } 4n_1 - n = \text{even}, \\ 1, 2, \dots, \frac{4n_1-n+1}{2}, & \text{if } 4n_1 - n = \text{odd}, \end{cases}$$

where

$$S_1(t_i) = \sum_{j=1}^{2n_1+2} \binom{2n_1+2}{j} (-1)^{j+1} G(t_{i-j}).$$

Since, by (11),  $G(t_i) = g_i$  for  $i = 1, 2, \dots, 2n_1 + 1$ , if we replace  $t_i$  by  $-t_i$ , then

$$S_1(-t_i) = \sum_{j=1}^{2n_1+2} \binom{2n_1+2}{j} (-1)^{j+1} g_{-i+j}.$$

Hence we get the constraints at the left end as

$$\sum_{j=0}^{2n_1+2} \binom{2n_1+2}{j} (-1)^j g_{-i+j} = 0, \tag{31}$$

where

$$i = \begin{cases} 1, 2, \dots, \frac{4n_1-n}{2}, & \text{if } 4n_1 - n = \text{even,} \\ 1, 2, \dots, \frac{4n_1-n+1}{2}, & \text{if } 4n_1 - n = \text{odd.} \end{cases}$$

#### 4.2.2 The right end imposed constraints

For the right end imposed values,  $g_i = S_1(t_i)$ ,  $i = N + 1, N + 2, \dots, N + (2n_1 - 1)$ , and

$$S_1(t_i) = \sum_{j=1}^{2n_1+2} \binom{2n_1+2}{j} (-1)^{j+1} g_{i-j}.$$

Hence we get the constraints at the right end as

$$\sum_{j=0}^{2n_1+2} \binom{2n_1+2}{j} (-1)^j g_{i-j} = 0, \tag{32}$$

where

$$i = \begin{cases} N + 1, N + 2, \dots, N + \frac{4n_1-n}{2}, & \text{if } 4n_1 - n = \text{even,} \\ N + 1, N + 2, \dots, N + \frac{4n_1-n-1}{2}, & \text{if } 4n_1 - n = \text{odd.} \end{cases}$$

### 5 The stable systems and their convergence

In this section, we present the linear and nonlinear stable systems of equations for the problems (3) and (4), respectively.

#### 5.1 The linear stable system

Since the system (17) is unstable, by combining  $4n_1 - n$  constraints, we get the stable system of the form

$$B_L G_L = D_L, \tag{33}$$

where the matrix  $B_L$  is defined as

$$B_L = ((\mathbb{B}_L)^T, (A_L)^T, (\mathbb{B}_R)^T)^T, \tag{34}$$

$G_L$  is defined in (19), and  $D_L$  is the matrix of order  $(N + 4n_1 + 1)$  defined as:

- If  $n$  is even,

$$D_L = (0, \dots, 0, 0, u^{n-1}(a), u^{n-2}(a), \dots, u'(a), u(a), d_L^T, u(b), u'(b), \dots, u^{n-2}(b), u^{n-1}(b), 0, 0, \dots, 0)^T; \tag{35}$$

- If  $n$  is odd,

$$\left\{ \begin{array}{l} D_L = (0, \dots, 0, 0, u^{n-2}(a), u^{n-3}(a), \dots, u'(a), u(a), d_L^T, u(b), \\ \quad u'(b), \dots, u^{n-1}(b), 0, 0, \dots, 0)^T, \\ \text{or} \\ D_L = (0, \dots, 0, 0, u^{n-1}(a), \dots, u'(a), u(a), d_L^T, u(b), \\ \quad u'(b), \dots, u^{n-3}(b), u^{n-2}(b), 0, 0, \dots, 0)^T, \end{array} \right. \tag{36}$$

where  $d_L$  is defined in (20). The matrix  $A_L$  is defined in (18), and the matrices  $\mathbb{B}_L$  and  $\mathbb{B}_R$  are of order  $(\frac{4n_1}{2} \times (N + 4n_1 + 1))$  constructed as follows:

In matrix  $\mathbb{B}_L$ ,

- If  $(4n_1 - n)$  or  $n$  is even, the first  $\frac{4n_1-n}{2}$  rows are constructed by using (31) for  $i = \frac{4n_1-n}{2}, \frac{4n_1-n}{2} - 1, \dots, 2, 1$ , respectively. The last  $\frac{n}{2}$  rows are obtained from  $u^n(0), u^{n-1}(0), \dots, u'(0), u(0)$ , respectively.
- If  $(4n_1 - n)$  or  $n$  is odd, the first  $\frac{4n_1-n+1}{2}$  rows are constructed by using (31)  $i = \frac{4n_1-n+1}{2}, \frac{4n_1-n+1}{2} - 1, \dots, 2, 1$ , respectively. The last  $\frac{n}{2}$  rows are obtained from  $u^n(0), u^{n-1}(0), \dots, u'(0), u(0)$ , respectively.

The construction of  $\mathbb{B}_R$  is as follows:

- If  $(4n_1 - n)$  or  $n$  is even, then the first  $\frac{n}{2}$  rows are obtained from  $u(1), u'(1), \dots, u^{n-1}(1), u^n(1)$ , respectively. The last  $\frac{4n_1-n}{2}$  rows are constructed by using (32) for  $i = N + 1, N + 2, \dots, N + \frac{4n_1-n}{2}$ , respectively.
- If  $(4n_1 - n)$  or  $n$  is odd, then the first  $\frac{n}{2}$  rows are obtained from  $u(1), u'(1), \dots, u^{n-1}(1), u^n(1)$ , respectively. The last  $\frac{4n_1-n-1}{2}$  rows are constructed by using (32) for  $i = N + 1, N + 2, \dots, N + \frac{4n_1-n-1}{2}$ , respectively.

If derivative constraints are given, then first approximate them with the help of (29) and (30) before using them.

### 5.2 The nonlinear stable system

Since the system (25) is unstable, by combining  $4n_1 - n$  imposed and  $n$  given constraints, we get a stable system of the form

$$B_{NL}G_{NL} = D_{NL}(g), \tag{37}$$

where the matrix  $B_{NL}$  is defined as

$$B_{NL} = ((\mathbb{B}_L)^T, (A_{NL})^T, (\mathbb{B}_R)^T)^T, \tag{38}$$

$G_{NL}$  is defined in (27), and  $D_{NL}(g)$  is the matrix of order  $(N + 4n_1 + 1)$  defined as:

- If  $n$  is even,

$$\begin{aligned} D_{NL} = & (0, \dots, 0, 0, u^{n-1}(a), u^{n-2}(a), \dots, u'(a), u(a), d_{NL}^T, u(b), \\ & u'(b), \dots, u^{n-2}(b), u^{n-1}(b), 0, 0, \dots, 0)^T; \end{aligned} \tag{39}$$

- If  $n$  is odd,

$$\begin{aligned}
 D_{NL} &= (0, \dots, 0, 0, u^{n-2}(a), u^{n-3}(a), \dots, u'(a), u(a), d_{NL}^T, u(b), \\
 &\quad u'(b), \dots, u^{n-1}(b), 0, 0, \dots, 0)^T, \\
 \text{or} \\
 D_L &= (0, \dots, 0, 0, u^{n-1}(a), \dots, u'(a), u(a), d_{NL}^T, u(b), \\
 &\quad u'(b), \dots, u^{n-3}(b), u^{n-2}(b), 0, 0, \dots, 0)^T, \tag{40}
 \end{aligned}$$

where  $d_{NL}$  is defined in (28). The matrix  $A_{NL}$  is defined in (26), and the matrices  $\mathbb{B}_L$  and  $\mathbb{B}_R$  of order  $(\frac{4n_1}{2} \times (N + 4n_1 + 1))$  are same as in the case of a stable linear system.

### 5.3 Existence of the solution

The coefficient matrices  $B_L$  and  $B_{NL}$  of the linear and nonlinear stable systems are banded and nonsingular. Remember that these matrices are not symmetric or diagonally dominant, though it can be proved that  $B_L$  and  $B_{NL}$  are nonsingular/invertible. If we ignore the first and last few rows and columns then these are symmetric matrices. Now consider the square symmetric part of  $B_L$  and asymmetric matrix  $B_{NL}$  of order  $(N + 1)$  given by

$$\begin{aligned}
 B_L^1 &= \begin{pmatrix} \rho_0^n & \rho_1^n & \rho_2^n & \rho_3^n & \dots & \rho_{2n_1}^n & \dots & 0 & 0 & 0 \\ \rho_{-1}^n & \rho_0^n & \rho_1^n & \rho_2^n & \dots & \rho_{2n_1-1}^n & \dots & 0 & 0 & 0 \\ \rho_{-2}^n & \rho_{-1}^n & \rho_0^n & \rho_1^n & \dots & \rho_{2n_1-2}^n & \dots & 0 & 0 & 0 \\ \rho_{-3}^n & \rho_{-2}^n & \rho_{-1}^n & \rho_0^n & \dots & \rho_{2n_1-3}^n & \dots & 0 & 0 & 0 \\ \rho_{-4}^n & \rho_{-3}^n & \rho_{-2}^n & \rho_{-1}^n & \dots & \rho_{2n_1-4}^n & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 & \dots & \rho_1^n & \rho_2^n & \rho_3^n \\ 0 & 0 & 0 & 0 & \dots & 0 & \dots & \rho_0^n & \rho_1^n & \rho_2^n \\ 0 & 0 & 0 & 0 & \dots & 0 & \dots & \rho_{-1}^n & \rho_0^n & \rho_1^n \\ 0 & 0 & 0 & 0 & \dots & 0 & \dots & \rho_{-2}^n & \rho_{-1}^n & \rho_0^n \end{pmatrix}, \\
 B_{NL}^2 &= \begin{pmatrix} \rho_1^n & \rho_2^n & \rho_3^n & \dots & \rho_{2n_1}^n & \dots & 0 & 0 & 0 \\ \rho_0^n & \rho_1^n & \rho_2^n & \dots & \rho_{2n_1-1}^n & \dots & 0 & 0 & 0 \\ \rho_{-1}^n & \rho_0^n & \rho_1^n & \dots & \rho_{2n_1-2}^n & \dots & 0 & 0 & 0 \\ \rho_{-2}^n & \rho_{-1}^n & \rho_0^n & \dots & \rho_{2n_1-3}^n & \dots & 0 & 0 & 0 \\ \rho_{-3}^n & \rho_{-2}^n & \rho_{-1}^n & \dots & \rho_{2n_1-4}^n & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & \dots & \rho_1^n & \rho_2^n & \rho_3^n \\ 0 & 0 & 0 & \dots & 0 & \dots & \rho_0^n & \rho_1^n & \rho_2^n \\ 0 & 0 & 0 & \dots & 0 & \dots & \rho_{-1}^n & \rho_0^n & \rho_1^n \\ 0 & 0 & 0 & \dots & 0 & \dots & \rho_{-2}^n & \rho_{-1}^n & \rho_0^n \end{pmatrix}.
 \end{aligned}$$

It can be shown that if  $a(t) > 0$  for  $0 \leq t \leq 1$ ,  $B_L^1$  and  $B_{NL}^2$  are always nonsingular and, for large  $N$ , matrices  $B_L$  and  $B_{NL}$  are very similar to  $B_L^1$  and  $B_{NL}^2$ , respectively. Since these are banded matrices, by the results of Kilic and Stanica [9], their inverses exist by LU factorization.

### 5.4 The solutions of linear and nonlinear systems

Now, we discuss the methods to find the solutions of the systems (33) and (37).

#### 5.4.1 The solution of linear system

The linear system of equations is defined in (33). We solve this system of equations by using Gaussian elimination method.

#### 5.4.2 The solution of nonlinear system

For the solution of nonlinear system (37), we do a few steps: First of all, we solve the following linear system with initial approximation  $G_{NL}^0$ :

$$B_{NL}G_{NL}^0 = D_{NL}^0, \tag{41}$$

where

$$\begin{cases} D_{NL}^0 = \{0, \dots, 0, 0, u^{n-1}(a), u^{n-2}(a), \dots, u'(a), u(a), F_0, F_1, \dots, F_N, u(b), \\ \quad u'(b), \dots, u^{n-2}(b), u^{n-1}(b), 0, 0, \dots, 0\}, \\ F_i = h^3 f(t_i, M_i, S), \quad i = 0, 1, 2, \dots, N, a \leq t \leq b, \\ M_i = u(a) + ih\left(\frac{u(b)-u(a)}{b-a}\right), \\ S = u(b) - u(a). \end{cases}$$

The solution of this system by Gaussian elimination method gives the initial approximation of the following nonlinear system:

$$B_{NL}G_{NL}^{m+1} = D_{NL}(G_{NL}^m), \quad m = 0, 1, 2, 3, \dots \tag{42}$$

Now continue the iterations by Gaussian elimination until

$$\|G_{NL}^m - G_{NL}^{m-1}\| \leq tol\epsilon, \tag{43}$$

where  $tol\epsilon$  is a chosen value. For example, someone can choose  $tol\epsilon = 10^{-6}$ .

## 6 The special cases of our algorithms

Here we present several special cases of our algorithms. We see that the algorithms based on interpolating and approximating schemes for solving linear and nonlinear DEs with the set of constraints are special cases of our algorithms.

### 6.1 The special cases of our algorithms based on interpolating schemes

Here we see that a number of algorithms based on the refinement schemes for solving differential equations are the special cases of our algorithms.

- If we take  $n = 2$ , the problem (3) with (5) at  $a = 0$  and  $b = 1$  becomes a second-order linear DE. For its solution, if we put  $n = n_1 = 2$  in (1), (2), (11), (31), (32), and (33) then we get the algorithms of Qu and Agarwal [2] and Mustafa et al. [10].
- If we take  $n = 2$ , the problem (4) with (5) at  $a = 0$  and  $b = 1$  becomes a second-order nonlinear DE. If we put  $n = n_1 = 2$  in (1), (2), (11), (31), (32), and (37) then we get the algorithm of Qu and Agarwal [3].

- If we take  $n = 3$ , the problem (3) and (6) at  $a = 0$  and  $b = 1$  becomes a third-order linear DE. If we put  $n = n_1 = 3$  in (1), (2), (11), (31), (32), and (33) then we get the algorithm of Mustafa and Ejaz [4, 11].
- If we take  $n = 3$ , the problem (4) and (6) at  $a = 0$  and  $b = 1$  becomes a third-order nonlinear DE. If we put  $n = n_1 = 3$  in (1), (2), (11), (31), (32), and (37) then we get the algorithm of Ejaz and Mustafa [6].
- If we take  $n = 4$ , the problem (3) and (5) at  $a = 0$  and  $b = 1$  becomes a fourth-order linear DE. If we put  $n = n_1 = 4$  in (1), (2), (11), (31), (32), and (33) then we get the algorithm of Ejaz et al. [5].
- If we take  $n = 4$ , the problem (4) and (5) at  $a = 0$  and  $b = 1$  becomes a fourth-order linear DE. If we put  $n = n_1 = 4$  in (1), (2), (11), (31), (32), and (37) then we get the algorithms of Mustafa et al. [7] and Ejaz et al. [12].

## 6.2 The special cases of our algorithms based on approximating schemes

The algorithms presented in Sects. 3 and 4 are based on the interpolating refinement schemes. Such algorithms can be restructured to get the algorithms based on approximating refinement schemes.

For the achievement of the purpose, we first choose an appropriate  $C^n$  approximating refinement scheme with complexity  $m_1 = 2, 4, 6, \dots$  and which satisfies (10). Then we discuss its properties as we have done for interpolating refinement schemes which are given in (7), (8), and (9). Proposition 1 will be stated in a similar way for this case. The algorithms for the solutions of the problems (3)–(6) by approximating refinement schemes can be obtained by replacing  $n_1 = \frac{m_1 - 2}{2}$  in the algorithms defined in Sects. 3 and 4.

Here we see that all the existing algorithms based on the approximating refinement schemes for solving differential equations will be special cases of our algorithms.

- If we take  $n = 2$ , the problem (3) and (5) at  $a = 0$  and  $b = 1$  becomes a second-order linear DE. Then select a suitable ( $m_1 = 6$ )-point approximating refinement scheme which satisfies (10). Further by substituting  $m_1 = 6$  or  $n_1 = 2$  in (11), (31), (32), and (33), we get the algorithm of Kanwal et al. [13].
- If we take  $n = 3$ , the problem (3) and (5) at  $a = 0$  and  $b = 1$  becomes a second-order linear DE. Then select a suitable ( $m_1 = 8$ )-point approximating refinement scheme which satisfies (10). Further by substituting  $m_1 = 8$  or  $n_1 = 3$  in (11), (31), (32), and (37), we get the algorithm of Manan et al. [14].
- If we take  $n = 4$ , the problem (3) and (5) at  $a = 0$  and  $b = 1$  becomes a fourth-order linear DE. Then select a suitable ( $m_1 = 10$ )-point approximating refinement scheme [15]. Further by substituting  $m_1 = 10$  or  $n_4 = 4$  in (11), (31), (32), and (33), we get the algorithm of Ejaz et al. [5].

## 7 Conclusion

In this paper, we first presented the generalized algorithms based on binary interpolating refinement schemes for the solution of the  $n$ th order linear and nonlinear differential equations with a set of constraints. Then we restructured these algorithms to get the algorithms based on approximating schemes. So, all the subdivision-based algorithms are easily restructured by substituting the suitable values of  $n$  and  $m$  in generalized algorithms for the solution of the  $n$ th order linear and nonlinear differential equations with a set of

constraints. Hence, we showed that several algorithms based on the interpolating and approximating refinement schemes for solving differential equations are the special cases of our unified algorithms.

## 8 Limitations of generalized algorithms

In this section, we present limitations of our generalized algorithms based on binary interpolating refinement schemes for the solution of the  $n$ th order linear and nonlinear differential equations with a set of constraints.

- Our algorithm is a generalization of all the existing algorithms based on interpolating or approximating subdivision schemes.
- We can easily reconstruct an algorithm to find the solution of any order linear and nonlinear differential equations with a set of constraints just by substituting the suitable values of  $n$  (order of DEs) and  $m_1$  (number of points in a subdivision scheme) in generalized algorithms.
- Our generalized algorithms reconstruct all the subdivision-based algorithms for the solution of the  $n$ th order linear and nonlinear ordinary differential equations with a set of constraints defined at  $a = 0$  and  $b = 1$ .
- Our generalized algorithm is not applicable when the constraints are defined at points other than  $a = 0$  and  $b = 1$ .

### Acknowledgements

This work is partially supported by NRP (P. No. 3183).

### Funding

This research received no external funding.

### Availability of data and materials

The data that support the findings of this study are available within the article.

### Competing interests

The authors declare that they have no competing interests.

### Authors' contributions

Conceptualization, GM and STE; Formal analysis, DB and Y-MC; Methodology, GM and DB; Supervision, GM; Writing original draft, STE and GM; Writing, reviewing, and editing, STE and Y-MC. All authors read and approved the final manuscript.

### Author details

<sup>1</sup>Department of Mathematics, The Government Sadiq College Women University Bahawalpur, Bahawalpur, Pakistan.

<sup>2</sup>Department of Mathematics, The Islamia University of Bahawalpur, Bahawalpur, Pakistan. <sup>3</sup>Department of Mathematics, Cankaya University, Ankara 06530, Turkey. <sup>4</sup>Institute of Space Sciences, 077125 Magurele-Bucharest, Romania.

<sup>5</sup>Department of Mathematics, Huzhou University, Huzhou, 313000, P.R. China. <sup>6</sup>Hunan Provincial Key Laboratory of Mathematical Modeling and Analysis in Engineering, Changsha University of Science and Technology, Changsha, 410114, P.R. China.

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 6 July 2020 Accepted: 8 February 2021 Published online: 23 February 2021

## References

1. Qu, R.: Curve and surface interpolation by subdivision algorithms. *Comput.-Aided Draft. Des. Manuf.* **4**(2), 28–39 (1994)
2. Qu, R., Agarwal, R.P.: Solving two point boundary value problems by interpolatory subdivision algorithms. *Int. J. Comput. Math.* **60**, 279–294 (1996)
3. Qu, R., Agarwal, R.P.: An iterative scheme for solving nonlinear two point boundary value problems. *Int. J. Comput. Math.* **64**(3–4), 285–302 (1997)
4. Mustafa, G., Ejaz, S.T.: Numerical solution of two point boundary value problems by interpolating subdivision schemes. *Abstr. Appl. Anal.* **2014**, Article ID 721314 (2014)

5. Ejaz, S.T., Mustafa, G., Khan, F.: Subdivision schemes based collocation algorithms for solution of fourth order boundary value problems. *Math. Probl. Eng.* **2015**, Article ID 240138 (2015)
6. Ejaz, S.T., Mustafa, G.: A subdivision based iterative collocation algorithm for nonlinear third order boundary value problems. *Adv. Math. Phys.* **2016**, Article ID 5026504 (2016)
7. Mustafa, G., Abbas, M., Ejaz, S.T., Ismail, A.I.M., Khan, F.: A numerical approach based on subdivision schemes for solving non-linear fourth order boundary value problems. *J. Comput. Anal. Appl.* **23**(4), 607–623 (2017)
8. Deng, C., Ma, W.: A unified interpolatory subdivision scheme for quadrilateral meshes. *ACM Trans. Graph.* **32**(3), 1–11 (2013)
9. Kilic, E., Stanica, P.: The inverse of banded matrix. *J. Comput. Appl. Math.* **237**, 126–135 (2013)
10. Mustafa, G., Ejaz, S.T., Baleanu, D., Ghaffar, A., Nisar, K.S.: A subdivision-based approach for singularly perturbed boundary value problem. *Adv. Differ. Equ.* **2020**(1), 282 (2020)
11. Mustafa, G., Ejaz, S.T.: A subdivision collocation method for solving two point boundary value problems of order three. *J. Appl. Anal. Comput.* **7**(3), 942–956 (2017)
12. Ejaz, S.T., Baleanu, D., Mustafa, G., Malik, S., Chu, Y.M.: The numerical solution of fourth order nonlinear singularly perturbed boundary value problems via 10-point subdivision scheme based numerical algorithm. *AIP Adv.* **10**(9), 095322 (2020)
13. Kanwal, G., Ghaffar, A., Hafeezullah, M.M., Manan, S.A., Rizwan, M., Rahman, G.: Numerical solution of 2-point boundary value problem by subdivision scheme. *Commun. Math. Appl.* **10**(1), 19–29 (2019)
14. Manan, S.A., Ghaffar, A., Rizwan, M., Rahman, G., Kanwal, G.: A subdivision approach to the approximate solution of 3rd order boundary value problem. *Commun. Math. Appl.* **9**(40), 499–512 (2018)
15. Mustafa, G., Ashraf, P., Aslam, M.: Binary univariate dual and primal subdivision schemes. *SeMA J.* **65**(1), 23–35 (2014)

Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)

---