Variations in 2D and 3D Models by a New Family of Subdivision Schemes and Algorithms for its Analysis

Rabia Hameed,1 Ghulam Mustafa,2 and Rizwan Majeed3

1Department of Mathematics, The Government Sadiq College Women University Bahawalpur, Bahawalpur 63100, Pakistan
2Department of Mathematics, The Islamia University of Bahawalpur, Bahawalpur 63100, Pakistan
3Directorate of Information Technology, The Islamia University of Bahawalpur, Bahawalpur 63100, Pakistan

Correspondence should be addressed to Ghulam Mustafa; ghulam.mustafa@iub.edu.pk

Received 7 February 2022; Revised 13 April 2022; Accepted 25 April 2022; Published 1 June 2022

Copyright © 2022 Rabia Hameed et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A new family of combined subdivision schemes with one tension parameter is proposed by the interpolatory and approximating subdivision schemes. The displacement vectors between the points of interpolatory and approximating subdivision schemes provide the flexibility in designing the limit curves and surfaces. Therefore, the limit curves generated by the proposed subdivision schemes variate in between or around the approximating and interpolatory curves. We also design few analytical algorithms to study the properties of the proposed schemes theoretically. The efficiency of these algorithms is analyzed by calculating their time complexity. The graphical representations and graphical properties of the proposed schemes are also analyzed.

1. Introduction

A common problem in computer aided geometric design consists of drawing smooth curves and surfaces that either interpolate or approximate a given shape. Subdivision schemes are one of the most successful approaches in this area. Subdivision schemes are used to construct smooth curves and surfaces from a given set of control points through iterative refinements. Due to its clarity and simplicity, subdivision schemes have been esteemed in many fields such as image processing, computer graphics, and computer animation. Generally, subdivision schemes can be categorized as interpolatory and approximating subdivision schemes. Interpolatory schemes get better shape control while approximating schemes have better smoothness. The most important interpolatory 4-point binary subdivision scheme was proposed by Dyn et al. [1] which gives the curves with $C^1$ continuity. This scheme was extended to the 6-point binary interpolatory scheme by Weissman [2]. The 6-point scheme presented by Weissman [2] gives the limiting curves with $C^2$ continuity. Most of the approximating schemes were developed from splines. Two of the most famous approximating schemes are Chaikin’s subdivision scheme [3] and cubic B-spline subdivision scheme [4], which actually generate uniform quadratic and cubic B-spline curves with $C^1$ continuity and $C^2$ continuity, respectively. There also exists a class of parametric subdivision schemes, called combined subdivision schemes. Combined subdivision schemes can be regarded either as a member of the approximating schemes or of the interpolatory schemes according to the specific values assumed by the parameters. Pan et al. [5] presented a combined ternary 3-point relaxed subdivision scheme with three tension parameters. Novara and Romani [6] extended their technique to construct a 3-point relaxed ternary combined subdivision scheme with three tension parameters to generate curves up to $C^3$ continuity. Recently, Zhang et al. [7] also used similar technique to construct a 4-point relaxed ternary scheme with four tension parameters to generate the curves up to $C^4$ smoothness. The nice performance of combined subdivision schemes motivates us to the direction of constructing the binary combined subdivision scheme. Our contribution in this manner is presented in this research.
The remainder of this article is organized as follows: Section 2 deals with some basic definition and basic results. Section 3 provides the framework for constructing a family of combined binary subdivision schemes. In Section 4, we provide the analytical properties of the proposed family of schemes. Section 5 gives the graphical behaviors and graphical properties of the proposed family of schemes. Concluding remarks are presented in Section 6.

2. Preliminaries

In this section, we present some basic notations and definition relating to the results which we have used in this article.

A general compact form of linear, uniform, and stationary binary univariate subdivision scheme $S_{2}$ which maps a polygon $P_{k} = \{P_{k}^{i}, i \in \mathbb{Z}\}$ to a refined polygon $P_{k+1} = \{P_{k+1}^{i}, i \in \mathbb{Z}\}$ is defined as

$$
\sum_{j \in \mathbb{Z}} a_i - 2j P_{k+1}^{j}, i \in \mathbb{Z}.
$$

(1)

The symbol of the above subdivision scheme is given by the Laurent polynomial

$$
a(z) = \sum_{i \in \mathbb{Z}} a_i z^i, z \in \mathbb{C}\setminus\{0\},
$$

(2)

where $a = \{a_i, i \in \mathbb{Z}\}$ is called the mask of the subdivision scheme. The necessary condition for the binary subdivision scheme (1) to be convergent is that its mask satisfies the basic sum rule

$$
\sum_{j \in \mathbb{Z}} a_i - 2j = 1, i = 0, 1,
$$

(3)

which is equivalent to the following relation:

$$
a(-1) = 2 \text{ and,}
\quad a(-1) = 0.
$$

(4)

Definition 1. An algorithm is a procedure for solving a mathematical problem in a finite number of steps that frequently involves repetition of an operation.

Definition 2. In computer science, the time complexity is the computational complexity that describes the amount of time it takes to run an algorithm. It is denoted by $T(n)$, where $n$ is the input of the given algorithm. The time complexity indicates the number of times the operations are executed in an algorithm. The number of instructions executed by a program is affected by the size of the input and how their elements are arranged.

Definition 3. Step count method considers each and every step to the given algorithm for calculating the final time complexity of the given algorithm. So, each and every step in the given algorithm contributes to the final complexity of the given algorithm.

Definition 4. Worst-case time complexity denotes the longest running time $\mathcal{O}(n)$ of an algorithm given any input of size $n$. It gives an upper bound on time requirements.

Definition 5. Best-case time complexity denotes the shortest running time $\Omega(n)$ of an algorithm given any input of size $n$. It gives a lower bound on time requirements.

3. Family of the Combined Primal Schemes

In this section, we discuss the construction of the family of $(2n + 2)$-point binary relaxed schemes. These schemes are the binary schemes, so are the combination of two refinement rules. One of these rules, consists of the affine combination of $(2n + 1)$-point of level $k$ to insert a new point at level $k + 1$, is used to update the vertex of the given polygon; hence, it is called the vertex rule. The other refinement rule, consists of the affine combination of $(2n + 2)$-points of level $k$ to insert a new point at level $k + 1$, is used for subdividing each edge of the given polygon and so is called the edge rule. Before going to the construction process of the family of combined subdivision schemes, we discuss our motivation behind this research here.

3.1. Motivation and Contribution. In this article, we define a novel technique for constructing combined subdivision schemes by using interpolatory and approximating subdivision schemes with optimal performance in curves and surface fitting. Since the family of interpolatory schemes by Deslauriers and Dubuc [8] gives optimal performance in reproducing polynomials and the family of approximating B-spline schemes gives good performance in fitting curves with higher continuity, so we are motivated to unify them into a single family of schemes.

The $(2n + 2)$-point interpolatory subdivision schemes defined by Deslauriers and Dubuc [8] can reproduce polynomials up to degree $2n + 1$ and generate the limiting curves with $C^1$. Whereas, the $(2n + 2)$-point relaxed B-spline approximating schemes are $C^{\infty}$-continuous, while the degree of polynomial reproduction of these schemes is only one. Since the discussed two schemes are the primal schemes, hence each of them is the combination of one vertex and one edge rules. The vertex rule of interpolatory schemes keeps the initial vertices at every level of subdivision, while the vertex rule of the approximating schemes updates the initial vertices at each level of subdivision. Similarly, the edge rule of each scheme subdivides every edge of the given polygon into two edges at each level of subdivision. The limit curves generated by the interpolatory and approximating schemes after 4 subdivision levels are shown in Figure 1. In this article, we construct a family of combined subdivision schemes with one parameter which controls the shape of the limit curve.

We construct the new family of schemes by the family of interpolatory schemes defined by Deslauriers and Dubuc [8] and the family of approximating B-spline schemes. Therefore, the proposed family of schemes carry the properties of these two types of schemes and give optimal numerical and
mathematical results. From Figure 2, Figure 3, we can see the better performance of the proposed schemes than that of their parent schemes (schemes which are used in their construction). Figure 2 shows that if we choose the value of tension parameter for the proposed 4-point scheme between \(-2.5\) and \(-2\), then this scheme is good for fitting noisy data. Figure 3(c) shows that the curves fitted by the proposed 6-point scheme do not give the artifacts as presented by the 6-point scheme [8] in Figure 3(a). This figure also shows that the curve fitted by the proposed scheme preserve the shape of the initial polygon if we take value of tension parameter between \(-0.5\) and \(0\), while the curve fitted by the 6-point B-spline scheme does not have this ability, as shown in Figure 3(b).

Construction of the family of schemes is discussed in coming subsections in detail.

3.2. Framework for the Construction of Family of Subdivision Schemes. The family of \((2n+2)\)-point interpolatory subdivision schemes, constructed by Lagrange’s interpolatory polynomials presented by Deslauriers and Dubuc [8], is defined by two subdivision rules. The vertex rule is

\[
R^{k+1}_{2n+2} = R^k_{2n+2},
\]

and the edge rule is

\[
P^{k+1}_{2n+1,2n+2} = \frac{1}{2^{4n+1}} \sum_{j=-n}^{n} \binom{n+1}{j} \frac{(-1)^{j} (n+1)}{(2j+1)} \binom{2n+1}{n} \binom{2n+1}{n+j+1} p^k_{j+1,2n+2}.
\]  

(5)

The refinement rules of the family of \((2n+2)\)-point relaxed approximating schemes, i.e. degree-\((4n+1)\) binary B-spline subdivision schemes, are

\[
\begin{align*}
Q^{k+1}_{2n+2} & = \frac{1}{2^{4n+1}} \sum_{j=-n}^{n} \binom{n+2}{j} \binom{4n+2}{2j+1} p^k_{j+n,2n+2}, \\
Q^{k+1}_{2n+1,2n+2} & = \frac{1}{2^{4n+1}} \sum_{j=-n}^{n} \binom{n+2}{j} \binom{4n+2}{2j} p^k_{j+n,2n+2},
\end{align*}
\]

(6)

where \(n \in \mathbb{Z}^+\). Let us denote the displacement vectors from the degree-\((4n + 1)\) B-spline refinement points \(Q^{k+1}_{2n+2}\) to the refinement points \(R^{k+1}_{2n+2}\) of \((2n + 2)\)-point interpolatory subdivision scheme defined in (5) after one step of refinements by \(D^{k+1}_{2n+2}\) and \(D^{k+1}_{2n+1,2n+2}\), respectively. Hence, the displacement vectors are

\[
\begin{align*}
D^{k+1}_{2n+2} & = R^{k+1}_{2n+2} - Q^{k+1}_{2n+2}, \\
D^{k+1}_{2n+1,2n+2} & = Q^{k+1}_{2n+1,2n+2} - Q^{k+1}_{2n+2}.
\end{align*}
\]

(7)

Now, a new family of primal combined \((2n+2)\)-point subdivision schemes can be obtained by translating the points \(R^{k+1}_{2n+2}\) and \(Q^{k+1}_{2n+2}\), that are provided by the binary interpolatory subdivision schemes (5) to the new position according to the displacement vectors \(aD^{k+1}_{2n+2}\) and \(aD^{k+1}_{2n+1,2n+2}\), respectively.

\[
\begin{align*}
R^{k+1}_{2n+2} & = R^{k+1}_{2n+2} + aD^{k+1}_{2n+2}, \\
R^{k+1}_{2n+1,2n+2} & = R^{k+1}_{2n+1,2n+2} + aD^{k+1}_{2n+1,2n+2},
\end{align*}
\]

(8)

which can be further written as

\[
\begin{align*}
p^{k+1}_{2n+2} & = \frac{-\alpha}{2^{4n+1}} \sum_{j=-n}^{n} \binom{n+2}{j} \binom{4n+2}{2j+1} p^{k+1}_{j+n,2n+2} + \frac{1}{2^{4n+1}} \left( 2^{4n+1} (1 + \alpha) - \alpha \right) \\
\end{align*}
\]

\[
\begin{align*}
p^{k+1}_{2n+1,2n+2} & = \frac{1}{2^{4n+1}} \sum_{j=-n}^{n} \binom{n+2}{j} \binom{4n+2}{2j} \frac{(-1)^{j-n-1} (n+1)}{2j-n-1} \binom{2n+1}{n} \binom{2n+1}{n+j+1} + \alpha \frac{(-1)^{j-n-1} (n+1)}{2j-n-1} \
\end{align*}
\]

\[
\begin{align*}
\left( \binom{2n+1}{n} \binom{2n+1}{n+j+1} - \binom{4n+2}{2j} \right) p^{k+1}_{j+n,2n+2}.
\end{align*}
\]

(9)
The rules defined in (9) are the refinement rules of the proposed family of \((2n+2)\)-point combined schemes, which are equivalent to the following general form:

\[
\begin{align*}
P_{2i,2n+2}^{k+1} &= \sum_{j=0}^{2n} a_{j,2n+2} P_{i+j-n,2n+2}^k, \\
P_{2i+1,2n+2}^{k+1} &= \sum_{j=0}^{2n+1} a_{j,2n+2} P_{i+j-n,2n+2}^k.
\end{align*}
\]  

(10)

The mask symbol of the above scheme is

\[
a_{2n+2}(z) = \sum_{j=0}^{4n+2} a_{j,2n+2} z^j = \frac{(1+z)^{2n+2}}{2^{2n+1}} A_{2n+2}(z),
\]

(11)

with

\[
A_{2n+2}(z) = \sum_{j=0}^{2n} A_{j,2n+2} z^j,
\]

(12)

where \(a_{j,2n+2}: j = 0, 1, \ldots, 4n+2\) and \(A_{j,2n+2}: j = 0, 1, \ldots, 2n\) are the coefficients of \(z^j: j = 0, 1, \ldots, 4n+2\) and \(z^j: j = 0, 1, \ldots, 2n\), respectively.

Remark 1. If \(\alpha \to 0\), the family of proposed \((2n+2)\)-point schemes converges to the family of \((2n+2)\)-point interpolatory schemes proposed in [8]. Whereas if \(\alpha \to -1\), the family of proposed \((2n+2)\)-point schemes converges to the family of \((2n+2)\)-point relaxed approximating B-spline schemes of degree- \((4n+1)\).

3.3. The Geometrical and Mathematical Interpretation of the Family of Schemes. The family of proposed schemes (9) is obtained by moving the points \(R_{2i,2n+2}^{k+1}\) and \(R_{2i+1,2n+2}^{k+1}\) of the family of schemes (5) to the new position according to the displacement vectors \(aD_{2i,2n+2}^{k+1}\) and \(aD_{2i+1,2n+2}^{k+1}\), respectively. In order to make the construction of the family of schemes understandable, we explain the following steps by fixing \(n = 1\) and \(k = 0\).

Step 1. We take the points \(P_{i-2,4}^0 = P_{i-2}^k, P_{i-1,4}^0 = P_{i-1}^k, P_{i,4}^0 = P_{i}^k, P_{i+1,4}^0 = P_{i+1}^k, P_{i+2,4}^0 = P_{i+2}^k\), respectively, after 4 subdivision levels.

The displacement vectors \(R_{2i,2n+2}^{k+1}\) and \(R_{2i+1,2n+2}^{k+1}\) are the coefficients of \(z^j: j = 0, 1, \ldots, 4n+2\) and \(p_k\) as the initial control points, i.e. at zeroth level of subdivision. The points are shown by red bullets in Figure 4.

Step 2. Now, we calculate points \(R_{2i+1}^{k+1}\) and \(R_{2i+1}^{k+1}\), which are denoted by blue solid squares in Figure 4. We calculate fifteen points (blue solid squares) by applying the following refinement rules of the 4-point interpolatory scheme of Deslauriers and Dubuc [8] on the initial points

\[
\begin{align*}
R_{2i}^{k+1} &= R_{2i}^{k+1} = q_i^k, \\
R_{2i+1}^{k+1} &= R_{2i+1,4}^{k+1} = -\frac{1}{16} p_{i-1}^k + \frac{9}{16} p_i^k + \frac{9}{16} p_{i+1}^k - \frac{1}{16} p_{i+2}^k.
\end{align*}
\]

(13)

All the points inserted by the above subdivision scheme are shown in Figure 4; however, we restrict our next calculations to only two points \(R_{2i}^{k+1}\) and \(R_{2i+1}^{k+1}\). In this figure, blue polygon is the initial polygon obtained from the scheme (13).

Step 3. In this step, we calculate and restrict next calculations to only the points \(Q_{2i}^{k+1}\) and \(Q_{2i+1}^{k+1}\) denoted by black solid squares. Again, the fifteen points, denoted by black solid squares in Figure 4, are calculated by the refinement rules of the 4-point relaxed binary B-spline scheme of degree-5 which is defined as follows:

\[
\begin{align*}
Q_{2i}^{k+1} &= Q_{2i}^{k+1} = \frac{3}{16} p_{i-1}^k + \frac{1}{5} 16 p_i^k + \frac{3}{16} p_{i+1}^k, \\
Q_{2i+1}^{k+1} &= Q_{2i+1,4}^{k+1} = \frac{1}{32} p_{i-1}^k + \frac{1}{5} 32 p_i^k + \frac{1}{32} p_{i+1}^k + \frac{1}{32} p_{i+2}^k.
\end{align*}
\]

(14)

In this figure, black polygon is the initial polygon made by the above scheme.

Step 4. Now, we calculate the displacement vectors from the points defined in (14) to the points defined in (13). These vectors are \(D_{2i}^{k+1} = D_{2i}^{k+1} - Q_{2i}^{k+1}\) and \(D_{2i+1}^{k+1} = D_{2i+1}^{k+1} - Q_{2i+1}^{k+1}\) in Figure 4, the above...
displacement vectors and the other related vectors are shown by green vectors.

Step 5. In this step, we calculate vectors $\alpha D_{k+1}^{i+1}$ and $\alpha D_{k+1}^{i+1}$, where $\alpha$ is any real number that can change the magnitude or the direction or both of the vectors $D_{k+1}^{i+1}$ and $D_{k+1}^{i+1}$. In Figure 4, these vectors are denoted by magenta lines.

Step 6. The last step of our method is to translate the points obtained by the refinement rules in (13) to the new position by using displacement vectors $\alpha D_{k+1}^{i+1}$ and $\alpha D_{k+1}^{i+1}$. So, we get following refinement rules of the new combined 4-point relaxed scheme

$$
\begin{aligned}
P_{2i}^{k+1} & = P_{2i+1}^{k+1} = \frac{3}{16} \alpha P_{i+1}^{k} + \left( 1 + \frac{3}{8} \alpha \right) P_{i}^{k} - \frac{3}{16} \alpha P_{i-1}^{k}, \\
P_{2i+1}^{k+1} & = P_{2i+2}^{k+1} = \left( 1 + \frac{3}{32} \alpha \right) P_{i+1}^{k} + \left( \frac{9}{16} + \frac{3}{32} \alpha \right) P_{i}^{k} + \left( \frac{9}{16} + \frac{3}{32} \alpha \right) P_{i+1}^{k} - \left( \frac{1}{16} + \frac{3}{32} \alpha \right) P_{i+2}^{k}.
\end{aligned}
$$

In Figure 4, the points of the above proposed scheme are denoted by black solid circles.
4. Analysis of the Proposed Family of Subdivision Schemes

In this section, we prove the characteristics of the proposed subdivision schemes analytically. We prove few theorems and present various algorithms to check these properties.

4.1. Support. The support of a subdivision scheme represents the portion of the limit curve effected by the displacement of a single control point from its initial place. Now, we calculate the support of the proposed family of subdivision schemes.

Theorem 1. If $\phi_{2n+2}$ is the basic limit function of the family of subdivision schemes (9), then, its support is $supp(\phi_{2n+2}) = [- (2n + 1), 2(n + 1)]$.

Proof. To calculate support of the proposed family of subdivision schemes, we define a function which give zero value to all control points except one point, $P_0$, whose value is one.

$$\phi(i) = P^0_i = \begin{cases} 0 & \text{for } i = 0, \\ 1 & \text{for } i \neq 0. \end{cases}$$

Let $\phi_{2n+2} : \mathbb{R} \rightarrow \mathbb{R}$ be the basic limit function such that $\phi_{2n+2}(i/2^k) = P^k_i$, Moreover, $\phi(i) = \phi_{2n+2}(i)$ and $P^0_0 = P^0_{2n+2}$. Therefore, the support of basic limit function $\phi_{2n+2}$ is equal to the support of the subdivision scheme $S_{2n+2}$.

When we apply a proposed $(2n + 2)$-point relaxed scheme on the initial data $P^0_0$, then at first subdivision step, the nonzero vertices are

$$\phi_{2n+2}\left(-\frac{2^0(2n + 1)}{2^1}\right) = P^1_{-2^0(2n+1), 2n+2} + P^1_{-2^0(2n+1)+1, 2n+2} + \cdots + P^1_{-2^0(2n+1)-1, 2n+2},$$

$$P^1_{2^0(2n+1), 2n+2} = \phi_{2n+2}\left(\frac{2^0(2n + 1)}{2^1}\right).$$

At second subdivision step, the nonzero vertices are

$$\phi_{2n+2}\left(-\frac{2^0 + 2^1}{2^2}(2n + 1)\right) = P^2_{-2^0(2n+1) + 2^0 + 2^1(2n+1), 2n+2} + P^2_{-2^0(2n+1) + 1, 2n+2} + \cdots + P^2_{-2^0(2n+1) - 1, 2n+2},$$

$$P^2_{2^0(2n+1) + 2^0 + 2^1, 2n+2} = \phi_{2n+2}\left(\frac{2^0 + 2^1}{2^2}(2n + 1)\right).$$

At third subdivision step, the nonzero vertices are
\[
\phi_{2n+2} \left( \frac{-2^0 + 2^1 + 2^2}{2^3} (2n + 1) \right) = P^3_{[2^0 + 2^1 + 2^2]}(2n+1,2n+2) = \phi_{2n+2} \left( \frac{2^n + 2^1 + 2^2}{2^3} (2n + 1) \right).
\]

(19)

Similarly, at \( k \)-th subdivision step, the nonzero vertices are

\[
\phi_{2n+2} \left( \frac{-\sum_{j=0}^{k-1} 2^j (2n + 1)}{2^k} \right) = P^k_{\sum_{j=0}^{k-1} 2^j (2n+1),2n+2} = \phi_{2n+2} \left( \frac{\sum_{j=0}^{k-1} 2^j (2n + 1)}{2^k} \right).
\]

(20)

Hence, the support size of basic limit function is the difference between subscripts of the maximum nonzero vertex and the minimum nonzero vertex, i.e.

\[
\text{support size of } \phi_{2n+2} = \frac{\sum_{j=0}^{k-1} 2^j (2n + 1)}{2^k} - \frac{\sum_{j=0}^{k-1} 2^j (2n + 1)}{2^k}
\]

\[= 2(2n + 1) \sum_{j=0}^{k-1} \frac{1}{2^j} \]

(21)

Applying limit \( k \rightarrow \infty \), we get support size of \( \phi_{2n+2} = 2(2n + 1) \). Hence, \( \text{supp} \phi_{2n+2} = [- (2n + 1), (2n + 1)] \) is the support region of \( \phi_{2n+2} \).

4.2. Smoothness Analysis. Now, we calculate the order of parametric continuity of the proposed subdivision schemes by using the Laurent polynomial method. Detailed information about refinement rules, Laurent polynomials, and convergence of a subdivision scheme can be found in [9–11]. The continuity of the subdivision schemes can be analyzed by the following theorems.

**Theorem 2** (see [10]). A convergent subdivision scheme \( S_b \) corresponding to the symbol

\[
a(z) = \left( \frac{1 + z}{2z} \right)^n b(z),
\]

(22)
\[\sum_{j=0}^{2n} \binom{4n + 2}{2j + 1} = \sum_{j=0}^{2n+1} \frac{(-1)^{j-n}(n+1)}{2j - 2n - 1} \binom{2n + 1}{j} \binom{2n + 1}{n} = \sum_{j=0}^{2n+1} \frac{4n + 2}{2j} = 2^{4n+1}. \tag{26}\]

Hence,
\[\frac{2n}{2^{4n+1}} \left(\frac{4n + 2}{2j + 1}\right) + (1 + \alpha) = 1, \tag{27}\]
and
\[\frac{2n}{2^{4n+1}} \left(\frac{4n + 2}{2j}\right) + (1 + \alpha) = 1. \tag{28}\]

Therefore, the family of subdivision schemes with the refinement rules defined in (9) satisfy the basic sum rule which is also the necessary condition for the convergence of a subdivision scheme.

We use Theorems 2 and 3 to design Algorithm 1. This algorithm is designed to calculate the level of parametric continuity of the limit curves generated by the proposed subdivision schemes. In Tables 1 and 2, we collect the results obtained from Algorithm 1 for \(L = 1\) and \(L = 2\), respectively. We use the step count method to find out the time complexity of the iterative Algorithm 1. Firstly, we compute the if-case time complexity.

Time complexity of Steps 2–15 of Algorithm 1 (if-case).

Hence, the if-case time complexity of Algorithm 1 is
\[T_f(n, L) = C_1 + C_2 + 2C_3 + 2C_4 + 2C_7 + C_8 + 2C_{10} + C_13 + C_{14} + 2^{L+1}C_5 + 2^{L+1}C_6 + 4nC_3 + 4nC_4 + 4nC_7 + 4nC_9 + 4nC_{10} + 4n^2C_5 + 4n^2C_6, \tag{29}\]
where \(C_1 = C_1 + C_2 + 2C_3 + 2C_4 + 2C_7 + C_8 + 2C_9 + 2C_{10} + C_{13} + C_{14}, \quad \bar{C}_2 = C_2 + C_3, \quad \bar{C}_3 = C_3 + C_4 + C_5 + C_6 + C_{10}, \quad \bar{C}_4 = C_4 + C_5 + C_6.\)

Now, we compute the else-case time complexity of Algorithm 1.

\[T_e(n, L) = C_{15} + C_{16} + 2C_{17} + 2C_{18} + C_{21} + 2C_{22} + 2C_{23} + C_{24} + 2nC_{17} + 2nC_{18} + 2nC_{22} + 2nC_{23} + 2^{L+1}C_{19} + 2^{L+1}C_{20} + 2n^2C_{19} + 2n^2C_{20}, \tag{30}\]
where \(\bar{C}_5 = C_{15} + C_{16} + C_{17} + C_{18} + C_{21} + 2C_{22} + 2C_{23} + C_{24}, \quad \bar{C}_6 = C_{17} + C_{18} + C_{22} + C_{23}, \quad \bar{C}_7 = C_{19} + C_{20}, \quad \bar{C}_8 = C_{19} + C_{20}.\)

4.3 Degrees of Polynomial Generation and Polynomial Reproduction. Generation and reproduction degrees are used to examine the behaviors of a subdivision scheme when the original data points lie on the curve of a
If the control points of the limit curve lie on curve of the same polynomial, then we say that the subdivision scheme reproduces polynomials of degree $d$. Mathematically, let $\Pi_d$ denote the space of polynomials of degree $d$ and $g, h \in \Pi_d$, an approximation operator $O$ generates polynomials of
degree $d$ if $Og = hh'g$, $h \in \Pi_d$, whereas $O$ reproduces polynomials of degree $d$ if $Og = gg'g \in \Pi_d$. Furthermore, the generation degree of a subdivision scheme is the maximum degree of polynomials that can theoretically be generated by the scheme, provided that the initial data is taken correctly. Evidently, it is not less than the reproduction degree. The readers may consult [9, 12] to find out more details about polynomial generation and polynomial reproduction of a subdivision scheme. We design

Table 3: Time complexity of Steps 2–15 of Algorithm 1 (if-case).

| Steps | Cost | Frequency | Total cost |
|-------|------|-----------|------------|
| 2     | $C_1$ | 1         | $C_1$      |
| 3     | $C_2$ | 1         | $C_2$      |
| 4     | $C_3$ | $4n + 2$  | $C_3 (4n + 2)$ |
| 5     | $C_4$ | $4n + 2$  | $C_4 (4n + 2)$ |
| 6     | $C_5$ | $2^k (4n + 2)$ | $C_5 (2^k (4n + 2))$ |
| 7     | $C_6$ | $2^k (4n + 2)$ | $C_6 (2^k (4n + 2))$ |
| 8     | $C_7$ | $4n + 2$  | $C_7 (4n + 2)$ |
| 9     | $C_8$ | 1         | $C_8$      |
| 10    | $C_9$ | $4n + 2$  | $C_9 (4n + 2)$ |
| 11    | $C_{10}$ | $4n + 2$ | $C_{10} (4n + 2)$ |
| 12    | $C_{11}$ | 1       | $C_{11}$ |
| 13    | $C_{12}$ | 1       | $C_{12}$ |
| 14    | $C_{13}$ | 1       | $C_{13}$ |
| 15    | $C_{14}$ | 1       | $C_{14}$ |

Table 4: Time complexity of Steps 16–25 of Algorithm 1 (else-case).

| Steps | Cost | Frequency | Total cost |
|-------|------|-----------|------------|
| 16    | $C_{15}$ | 1         | $C_{15}$ |
| 17    | $C_{16}$ | 1         | $C_{16}$ |
| 18    | $C_{17}$ | $2n + 2$  | $C_{17} (2n + 2)$ |
| 19    | $C_{18}$ | $2n + 2$  | $C_{18} (2n + 2)$ |
| 20    | $C_{19}$ | $2^k (2n + 2)$ | $C_{19} (2^k (2n + 2))$ |
| 21    | $C_{20}$ | $2^k (2n + 2)$ | $C_{20} (2^k (2n + 2))$ |
| 22    | $C_{21}$ | 1         | $C_{21}$ |
| 23    | $C_{22}$ | $2n + 2$  | $C_{22} (2n + 2)$ |
| 24    | $C_{23}$ | $2n + 2$  | $C_{23} (2n + 2)$ |
| 25    | $C_{24}$ | 1         | $C_{24}$ |

Algorithm 2 and Algorithm 3, based on the results given in [12], to check the degrees of polynomial generation and polynomial reproduction of proposed schemes, respectively. Outputs obtained by implementing these algorithms are summarized in Tables 7 and 8, respectively.

Now, we find the time complexity of Algorithm 2. Firstly, we compute the if-case (Steps 2–11) time complexity and then we find the else-case time complexity (Steps 12–22). Hence, the worst-case time complexity will be the time complexity of Algorithm 2.

Time complexity of Steps 2–11 of Algorithm 2 (if-case).

Table 5: Time complexity of Steps 2–11 of Algorithm 2 (if-case).

| Steps | Cost | Frequency | Total cost |
|-------|------|-----------|------------|
| 2     | $C_1$ | 1         | $C_1$      |
| 3     | $C_2$ | 1         | $C_2$      |
| 4     | $C_3$ | $4n + 2$  | $C_3 (4n + 2)$ |
| 5     | $C_4$ | $4n + 2$  | $C_4 (4n + 2)$ |
| 6     | $C_5$ | $2^k (4n + 2)$ | $C_5 (2^k (4n + 2))$ |
| 7     | $C_6$ | $2^k (4n + 2)$ | $C_6 (2^k (4n + 2))$ |
| 8     | $C_7$ | $4n + 2$  | $C_7 (4n + 2)$ |
| 9     | $C_8$ | 1         | $C_8$      |
| 10    | $C_9$ | $4n + 2$  | $C_9 (4n + 2)$ |
| 11    | $C_{10}$ | $4n + 2$ | $C_{10} (4n + 2)$ |
| 12    | $C_{11}$ | 1       | $C_{11}$ |
| 13    | $C_{12}$ | 1       | $C_{12}$ |
| 14    | $C_{13}$ | 1       | $C_{13}$ |
| 15    | $C_{14}$ | 1       | $C_{14}$ |

Table 6: Time complexity of Steps 12–21 of Algorithm 2 (else-case).

| Steps | Cost | Frequency | Total cost |
|-------|------|-----------|------------|
| 12    | $C_{11}$ | 1         | $C_{11}$ |
| 13    | $C_{12}$ | 1         | $C_{12}$ |
| 14    | $C_{13}$ | $2n + 2$  | $C_{13} (2n + 2)$ |
| 15    | $C_{14}$ | $2n + 2$  | $C_{14} (2n + 2)$ |
| 16    | $C_{15}$ | $2n + 2$  | $C_{15} (2n + 2)$ |
| 17    | $C_{16}$ | $2n + 2$  | $C_{16} (2n + 2)$ |
| 18    | $C_{17}$ | 1         | $C_{17}$ |
| 19    | $C_{18}$ | 1         | $C_{18}$ |
| 20    | $C_{19}$ | 1         | $C_{19}$ |
| 21    | $C_{20}$ | 1         | $C_{20}$ |
and the worst-case which is Steps 6–7. Hence, the if-case time complexity of Algorithm 2 is

\[ T_f(n) = C_1 + C_2 + 2C_3 + 2C_4 + 2C_5 + 2C_6 + C_9 + C_{10} + 4nC_3 + 4nC_4 + 4nC_5 + 4nC_6, \]

where \( C_1 = C_3 + C_4 + 2C_6 + C_9 + C_{10} \) and \( C_2 = C_3 + C_4 + C_5 + C_6 \). Now, we compute the else-case time complexity (Steps 12–21).

Time complexity of Steps 12–21 of Algorithm 2 (else-case).

To calculate time complexity of Steps 12–21, we ignore total cost of Steps 18–19. Therefore, the else-case time complexity of Algorithm 2 is

\[ T_e(n) = C_{11} + C_{12} + 2C_{13} + 2C_{14} + 2C_{15} + 2C_{16} + C_{19} + C_{20} + 2nC_{13} + 2nC_{14} + 2nC_{15} + 2nC_{16} = C_3 + 2nC_4 \in \Theta(n), \]

(32)

where \( C_3 = C_{11} + C_{12} + 2C_{13} + 2C_{14} + 2C_{15} + 2C_{16} + C_{17} + C_{18} + C_{19} + C_{20} \) and \( C_4 = C_{13} + C_{14} + C_{15} + C_{16} \). Hence, time complexity of Algorithm 2 is \( \Theta(n) \). In the same manner, we can prove that the time complexity of Algorithm 3 is also \( \Theta(n) \).

4.4. Gibbs Oscillations. According to [13], linear subdivision schemes exhibit overshoots or undershoots Gibboscillations near to the points of discontinuity of a discontinuous function. The schemes that show undershoots generate curves

\[
\begin{array}{c|c|c|c|c}
 n & 1 & 2 & 3 & 4 \\
\hline
 Schemes & 4-point scheme (relaxed) & 6-point scheme (relaxed) & 8-point scheme (relaxed) & 10-point scheme (relaxed) \\
\hline
\text{Generation degree } \forall \alpha & 3 & 5 & 7 & 9 \\
\text{Generation degree } \alpha = -1 & 5 & 9 & 13 & 17 \\
\end{array}
\]
(1) input: The symbol of the proposed schemes, i.e.
\[ a_{2n+2}(z) = a_{2n+2} z^2 + a_{2n+2} z^2 + a_{2n+2} z^2 + \cdots + \]
\[ a_{4n+2,2n+2} z^{2n+1} + \cdots + A_{2n+2} z^2 + A_{2n+2} z^2 + \cdots + A_{2n+2} z^2, \]
\[ n \in \mathbb{Z} \quad \{ \}
\[ a_{2n+2} \cdot i = 0, 1, \ldots, 4n+2 \text{ are the coefficients (depends on } a \text{) of } z': i = 0, 1, \ldots, 4n+2 \text{ respectively in } a_{2n+2}(z)
\]
(2) calculate: \( \tau = a'(z)/2 \)
(3) for \( j = 0 \) to \( 4n+3 \) do
(4) calculate: \( A_j = [a_{2n+2}]_{z \rightarrow \tau}, B_j = \prod_{p=0}^{j-1} (\tau - p) \) and \( C_j = a_{2n+2}(z)|_{z=-1} \)
(5) if \( A_j = B_j \) then and \( C_j = 0 \)
(6) scheme reproduce polynomials of degree \( j \forall a \)
(7) else
(8) calculate: \( A_j, B_j \) and \( C_j \) for \( a = 0 \)
(9) if \( A_j = B_j \) and \( C_j = 0 \) for \( a = 0 \) then
(10) scheme reproduce polynomials of degree \( j \) for \( a = 0 \)
(11) else
(12) go to 15
(13) end if
(14) end if
(15) end for
(16) output: degree of reproduction of the schemes corresponding to \( a_{2n+2}(z) \) for all \( a \) or for \( a = 0 \)

**Algorithm 4:** Undershoot Gibbs phenomenon of the proposed schemes.

| \( n \quad \rightarrow \) | \( k \quad \rightarrow \) | 1 | 2 | 3 |
|----------------|------|----|----|----|
| Schemes | \( 4 \)-point scheme (relaxed) | \( 6 \)-point scheme (relaxed) | 8-point scheme (relaxed) |
| \( 0 \) | \( a < 0 \) | \( a < 0 \) | \( a < 0 \) |
| 1 | \(-5.444444444 < a < 0 \) | \(-4.013223140 < a < 0 \) | \(-3.523131552 < a < 0 \) |
| 2 | \(-9.494261050 < a < 0 \) | \(-6.840555349 < a < 0 \) | \(-5.821687339 < a < 0 \) |
| 3 | \(-5.618133906 < a < 0 \) | \(-5.583827202 < a < 0 \) | \(-7.639304050 < a < 0 \) |

(1) input: \( n \in \mathbb{Z}^+, k, P^0_0, i \in \mathbb{Z} \) (\( \rightarrow \) \( P^0_i \)) are taken from a discontinuous function, i.e.
\[
P^0_0 = \begin{cases} \alpha & \text{for } i \in [-n, n - q^1], \\ -10 & \text{for } i \in [0], t + n \in [0]. \end{cases}
\]
(2) for \( m = 0 \) to \( k \) do
(3) calculate: \( P^m_{2i,2} = -\alpha/2^{m+1} \sum_{j=0}^{2m} \left[ 4n + 2 \left( \frac{2^{m+1} + 1}{2n + 1} \right) \right] P^m_{i, n-2n+1} + 1/2^{m+1} \left[ 1 + \alpha \right] \left( 2n + 2 \right) \left( 2n + 1 \right) \]
\[
\text{\( P^m_{2i,2} = 1/2^{m+1} \sum_{j=0}^{2m} \left[ \left( \frac{2^{m+1} + 1}{2n + 1} \right) \left( \frac{2n + 1}{j} \right) \right] + \alpha \times \left( \frac{2n + 2}{2j} \right) \right] \)}
\]
(4) calculate: \( P^m_{2i,2n+2} = 1/2^{m+1} \sum_{j=0}^{2m} \left[ \left( -1 \right)^{j-n-1} (n+1)/2j - 2n - 1 \right] \left( \frac{2n + 1}{j} \right) \left( \frac{2n + 1}{j} \right) \) \( \text{\( \left( 4n + 2 \right) \left( 2n + 1 \right) \right] \) \)}
\]
(5) end for
(6) calculate: intervals of \( a \) for which \( P^0_{-1,2n+2} < S^0_{\xi} \) (\( P^0_{2i,2} \))
(7) calculate: intervals of \( a \) for which \( P^0_{0,2n+2} > S^0_{\xi} \) (\( P^0_0 \))
(8) calculate: intersection of intervals of \( a \) from Steps 6–7
(9) output: range of \( a \) for which the given scheme exhibits undershoot Gibbs phenomenon at specific level of subdivision \( k \).

**Algorithm 3:** Reproduction degree of the schemes.

(1) \input: \( n \in \mathbb{Z}^+, k, P^0_0, i \in \mathbb{Z} \) (\( \rightarrow \) \( P^0_i \)) are taken from a discontinuous function, i.e.
\[
P^0_0 = \begin{cases} \alpha & \text{for } i \in [-n, n - q^1], \\ -10 & \text{for } i \in [0], t + n \in [0]. \end{cases}
\]
(2) for \( m = 0 \) to \( k \) do
(3) calculate: \( P^m_{2i,2} = -\alpha/2^{m+1} \sum_{j=0}^{2m} \left[ 4n + 2 \left( \frac{2^{m+1} + 1}{2n + 1} \right) \right] P^m_{i, n-2n+1} + 1/2^{m+1} \left[ 1 + \alpha \right] \left( 2n + 2 \right) \left( 2n + 1 \right) \]
\[
\text{\( P^m_{2i,2} = 1/2^{m+1} \sum_{j=0}^{2m} \left[ \left( -1 \right)^{j-n-1} (n+1)/2j - 2n - 1 \right] \left( \frac{2n + 1}{j} \right) \left( \frac{2n + 1}{j} \right) \) \)}
\]
(4) calculate: \( P^m_{2i,2n+2} = 1/2^{m+1} \sum_{j=0}^{2m} \left[ \left( -1 \right)^{j-n-1} (n+1)/2j - 2n - 1 \right] \left( \frac{2n + 1}{j} \right) \left( \frac{2n + 1}{j} \right) \) \( \text{\( \left( 4n + 2 \right) \left( 2n + 1 \right) \right] \) \)}
\]
(5) end for
(6) calculate: intervals of \( a \) for which \( P^0_{-1,2n+2} > S^0_{\xi} \) (\( P^0_{2i,2} \))
(7) calculate: intervals of \( a \) for which \( P^0_{0,2n+2} < S^0_{\xi} \) (\( P^0_0 \))
(8) calculate: intersection of intervals of \( a \) from Steps 6–7
(9) output: range of \( a \) for which the given scheme exhibits undershoot Gibbs phenomenon at specific level of subdivision \( k \).

free from oscillations while the schemes with overshoots give unpleasant oscillations in the limit curves. Let \( 0 \xi c \), and a function \( p \) is said to be discontinuous at a point \( \xi \), if
\[
\begin{align*}
\forall x \neq \xi, & \quad p(x) = p_+(x) \quad \text{with } p_+ \in C^\infty ([-\xi, \xi]), \\
\forall x \neq \xi, & \quad p(x) = p_-(x) \quad \text{with } p_- \in C^\infty ([\xi, \infty]),
\end{align*}
\]
Table 10: Implementation of Algorithm 5 for \( n = 1, 2, 3 \).

| \( n \) | 1 | 2 | 3 |
|-------|---|---|---|
| Schemes | 4-point scheme (relaxed) | 6-point scheme (relaxed) | 8-point scheme (relaxed) |
| \((\alpha_1, \alpha_2)\) | \((-2.666666667, -0.666666667)\) | \((-1.200000000, -0.5263157895)\) | \((-1.721008403, -0.9523809524)\) |
| \((\beta_1, \beta_2)\) | \((-1.555555556, -0.666666667)\) | \((-1.247058824, -0.5882352941)\) | \((-1.149842822, -0.6060606061)\) |
| \((\gamma_1, \gamma_2)\) | \((-1.555555556, -0.666666667)\) | \((-1.200000000, -0.5882352941)\) | \((-1.149842822, -0.9523809524)\) |

(1) **input:** The mask \( a_{j,2n+2} \): \( j = 0, 1, \ldots, 4n + 2 \) of the family of proposed schemes whose mask symbol is defined in (11), \( n \in \mathbb{Z}^+ \)
(2) set: \( A_j = a_{j,2n+2} \)
(3) for \( j = 0 \) to \( 4n + 2 \) do
(4) calculate: an interval, say \( (\alpha'_1, \alpha'_2) \), for \( \alpha \) by solving \( A_j > 0 \)
(5) end for
(6) calculate: intersection, say \( (\alpha_1, \alpha_2) \), of \( 4n + 3 \) intervals \( (\alpha'_1, \alpha'_2) \): \( j = 0, 1, \ldots, 4n + 2 \)
(7) for \( j = 0 \) to \( 2n \) do
(8) calculate: an interval, say \( (\beta'_1, \beta'_2) \), for \( \alpha \) by solving \( A_j < A_{j+1} \)
(9) end for
(10) calculate: intersection, say \( (\beta_1, \beta_2) \), of \( 2n + 1 \) intervals \( (\beta'_1, \beta'_2) \): \( j = 0, 1, \ldots, 2n \)
(11) calculate: intersection, say \( (\gamma_1, \gamma_2) \), of \( 2 \) intervals \( (\alpha_1, \alpha_2) \) and \( (\beta_1, \beta_2) \)
(12) **return** \( (\alpha_1, \alpha_2), (\beta_1, \beta_2), (\gamma_1, \gamma_2) \)
(13) **output:** the proposed \((2n+2)\)-point scheme has a bell-shaped mask for \( \alpha \in (\gamma_1, \gamma_2) \)

Algorithm 5: Bell-shaped mask of the proposed schemes.

Figure 5: Dashed lines are the initial polylines, while the solid lines show the limit curves generated by the proposed schemes at different values of parameter \( \alpha \). (a) 4-point scheme. (b) 6-point scheme. (c) 8-point scheme.

with \( p_\pm (\xi) > p_\pm (\zeta) \).

A scheme has the overshoots near to the point \( \xi \) of discontinuity if

\[
\forall \xi < \zeta, \quad p_\pm(\xi) > S^\infty_A(p^0)(\xi),
\]

\[
\forall \xi > \zeta, \quad p_\pm(\xi) < S^\infty_A(p^0)(\xi).
\]

A scheme has the undershoots near to the point \( \xi \) of discontinuity if

\[
\forall \xi > \zeta, \quad p_\pm(\xi) > S^\infty_A(p^0)(\xi),
\]

\[
\forall \xi < \zeta, \quad p_\pm(\xi) < S^\infty_A(p^0)(\xi).
\]

Algorithm 4 is designed to calculate the range of tension parameter \( \alpha \) for which the proposed schemes exhibit undershoot Gibbs oscillations. The results obtained from this algorithm are summarized in Table 9. We calculate the time complexity of Algorithm 4 which depends on the input
value $k$ by using the step count method. The time complexity of this algorithm does not depend on the input value $n$. Hence, the time complexity of this algorithm is $O(k)$.

4.5. Shape Preserving Properties. Here, we discuss the shape preserving properties, monotonicity and convexity preservation, of the proposed family of subdivision schemes. We prove that the proposed subdivision schemes preserve monotonicity and convexity for a special interval of tension parameter. For this, firstly, we calculate an interval of $\alpha$ for which the proposed subdivision schemes having the bell-shaped mask. Then, by using results given in [13], we prove that the proposed schemes preserve monotonicity and convexity for a special interval of $\alpha$.

A $2N$-degree polynomial $\sum_{j=0}^{2N} b_j x^j$ is said to be a polynomial with bell-shaped coefficients if it satisfies the following conditions:

\[
\begin{align*}
\text{supp}(b_j) &= [0,2N], \\
b_j &> 0 \text{ for } j \in [0,2N], \\
b_j = b_{2N-j} \text{ for } j \in [0,2N], \\
b_j < b_{j+1} \text{ for } j \in [0,N-1].
\end{align*}
\]
Figure 8: Curves generated by the proposed schemes after four subdivision steps where the initial data is taken from the convex function \( f(x) = \sqrt{x^2 + 2} \). (a) 4-point scheme. (b) 6-point scheme. (c) 8-point scheme.

Figure 9: Solid lines in (a), (b), and (c) show the limit curve generated by the 4-point, 6-point, and 8-point proposed schemes. (a) \( \alpha \approx -0.28 \), (b) \( \alpha \approx -0.25 \), and (c) \( \alpha \approx -0.19 \).

Figure 10: Solid lines in (a), (b), and (c) show the limit curve generated by the 4-point, 6-point, and 8-point proposed schemes. (a) \( \alpha \approx -2.5 \), (b) \( \alpha \approx -2 \), and (c) \( \alpha \approx -1.7 \).
Figure 11: Surfaces produced by the tensor product of proposed 4-point relaxed scheme after four subdivision levels. (a) Initial mesh, (b) $\alpha = -2.5$, (c) $\alpha = -2$, (d) $\alpha = -1.5$, (e), (f), (h), (i) $\alpha = -1$, (j) $\alpha = -0.5$, (k) $\alpha = 0$, (l) $\alpha = 0.25$, (m), (n), (o), (p).
The mask symbol of the proposed family of schemes is defined in (10) which is a polynomial of degree \((4n + 2)\) satisfying the following two conditions:

\[
\text{supp}(a_{2n+2}) = [0, 4n + 2], \\
a_{2n+2} = 4a_{4n+2} - a_0 	ext{ for } j \in [0, 4n + 2].
\]  

(37)

The other two conditions depend on value of \(\alpha\) which can be calculated by Algorithm 5. The outputs obtained from Algorithm 5 are given in Table 10.

**Theorem 5.** The proposed 4-point, 6-point, and 8-point subdivision schemes preserve monotonicity for \(-1.555555555 \leq \alpha < -0.666666667, -1.2000000 \leq \alpha < -0.5882352941, \) and \(-1.149842822 \leq \alpha < -0.9523809524), respectively.

**Proof.** By Theorem 4, we see that the proposed subdivision schemes satisfy the basic sum rule. Also from Algorithm 5, we get that the proposed 4-point, 6-point, and 8-point subdivision schemes have bell-shaped mask for \(-1.555555555 \leq \alpha < -0.666666667, -1.20000000 \leq \alpha < -0.5882352941, \) and \(-1.149842822 \leq \alpha < -0.9523809524), respectively. So it is easy to prove the required result by using ([13], Theorem 3).

**Theorem 6.** The proposed 4-point, 6-point, and 8-point subdivision schemes preserve convexity for \(-1.555555555 \leq \alpha < -0.666666667, -1.20000000 \leq \alpha \leq -0.5882352941, \) and \(-1.149842822 \leq \alpha < -0.9523809524), respectively.

**Proof.** By Theorem 4, we see that the proposed subdivision schemes satisfy the basic sum rule. Also from Algorithm 5, we get that the proposed 4-point, 6-point, and 8-point subdivision schemes have bell-shaped mask for \(-1.555555555 \leq \alpha < -0.666666667, -1.20000000 \leq \alpha < -0.5882352941, \) and \(-1.149842822 \leq \alpha < -0.9523809524), respectively. Furthermore from (11), we can also see that the mask symbols of the proposed subdivision schemes have the factor \((1 + \varepsilon)^2\). Hence, the required result is proved by ([13], Theorem 4).

The proposed Algorithm 5 needs \(\mathcal{O}(n)\) time to give output.

5. Numerical Examples

In this section, we show the numerical performance of the proposed schemes, based on the results which have proved in previous section. Figure 5 displays the support of the subdivision scheme when the initial data is taken from a basic limit function defined in (16). Figure 6 shows the curves fitted by the proposed \((2n + 2)\)-point subdivision schemes. It is clear from this figure that the tension parameter involved in the proposed subdivision schemes provides relaxation in drawing and designing different types of curves by using the same initial points. Figures 7 and 8 show the performance of the proposed subdivision schemes when the initial data is taken from monotonic and convex functions, respectively. Figures 9 and 10 show that the proposed subdivision schemes also carry the characteristics of keeping the shape of the initial polygon and of fitting the noisy data points, respectively, which is an important characteristic of the proposed family. In Literature, it is hard to find a single subdivision scheme which carries both characteristics simultaneously. The performance of the tensor product scheme of the proposed 4-point scheme in surface fitting is shown in Figure 11. Subfigures Figures 11(e), 11(f), 11(g), 11(h), 11(m), 11(n), 11(o), and 11(p), respectively are the mirror images in \(ij\)-plane of Subfigures 11(a), 11(b), 11(c), 11(d), 11(i), 11(j), 11(k), and 11(l).

6. Concluding Remarks

In this article, we have derived the general refinement rules of the family of \((2n + 2)\)-point primal binary schemes. The proposed schemes are originated from the schemes by Deslauriers and Dubuc [8]. We have also presented certain efficient algorithms with linear time complexity to check the properties of the proposed schemes. The beauty of the proposed family of schemes is that they carry the characteristics of both of its parent schemes. The proposed schemes show optimal continuity and polynomial generation. Proposed schemes reproduce polynomials up to degree one, while for specific value of tension parameters, they can reproduce polynomials up to degree \(2n + 1\). The proposed schemes eliminate Gibbs phenomenon for negative values of tension parameter. Moreover, for a specific range of tension parameter, the proposed schemes also preserve monotonicity and convexity. The proposed schemes provide flexibility in fitting curves and surfaces, that is, using the same data points, we can vary the 2D and 3D models in between or around the control polygon/mesh. Proposed schemes are a good choice for noisy data. In future, we will derive the nonuniform form of the proposed schemes.

Since from the methods used in our work, some of the most representative computational intelligence algorithms can be used to solve the problems, such as Monarch Butterfly Optimization [14, 15], Earthworm Optimization Algorithm [16], Elephant Herding Optimization [17], Moth Search Algorithm [18], SISGA Algorithm [19], Hunger Games Search [20], Runge Kutta Optimizer Method [21], Colony Predation Algorithm [22], and Harris Hawks Optimization Method [23]. These algorithms/methods can be used in the future to solve these type of problems. [24, 25].

Data Availability

The data used to support the findings of the study are available within this paper.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.
Authors’ Contributions

Rabia Hameed conceptualized the study and performed methodology; Ghulam Mustafa made formal analysis and written the original draft; Rizwan Majeed reviewed and edited the study, made visualization of shapes, and programmed the study.

References

[1] N. Dyn, D. Levin, and J. A. Gregory, “A 4-point interpolatory subdivision scheme for curve design,” Computer Aided Geometric Design, vol. 4, no. 4, pp. 257–268, 1987.
[2] A. Weissman, “A 6-point Interpolatory Subdivision Scheme for Curve Design,” (M. S. thesis, Tel-Aviv University, Tel Aviv, Israel, 1989.
[3] G. M. Chaikin, “An algorithm for high-speed curve generation,” Computer Graphics and Image Processing, vol. 3, no. 4, pp. 346–349, 1974.
[4] E. Cohen, T. Lyche, and R. Riesenfeld, “Discrete B-splines and subdivision techniques in computer-aided geometric design and computer graphics,” Computer Graphics and Image Processing, vol. 14, no. 2, pp. 87–111, 1980.
[5] J. Pan, S. Lin, and X. Luo, “A combined approximating and interpolating subdivision scheme with C2 continuity,” Applied Mathematics Letters, vol. 25, no. 12, pp. 2140–2146, 2012.
[6] P. Novara and L. Romani, “Complete characterization of the regions of C2 and C3 convergence of combined ternary 4-point subdivision schemes,” Applied Mathematics Letters, vol. 62, pp. 84–91, 2016.
[7] L. Zhang, H. Ma, S. Tang, and J. Tan, “A combined approximating and interpolating ternary 4-point subdivision scheme,” Journal of Computational and Applied Mathematics, vol. 349, pp. 563–578, 2019.
[8] G. Deslauriers and S. Dubuc, “Symmetric iterative interpolation processes,” Constructive Approximation, vol. 5, no. 1, pp. 49–68, 1989.
[9] M. Charina and C. Conti, “Polynomial reproduction of multivariate scalar subdivision schemes,” Journal of Computational and Applied Mathematics, vol. 240, pp. 51–61, 2013.
[10] N. Dyn, “Subdivision schemes in computer-aided geometric design,” Advances in Numerical Analysis, vol. 2, pp. 36–104, 1992.
[11] N. Dyn and D. Levin, “Subdivision schemes in geometric modelling,” Acta Numerica 2002, vol. 11, pp. 73–144, 2002.
[12] C. Conti and K. Hormann, “Polynomial reproduction for univariate subdivision schemes of any arity,” Journal of Approximation Theory, vol. 163, no. 4, pp. 413–437, 2011.
[13] R. Hameed and G. Mustafa, “Family of a -point b-ary subdivision schemes with bell-shaped mask,” Applied Mathematics and Computation, vol. 309, pp. 289–302, 2017.
[14] Y. Feng, X. Yu, and G. G. Wang, “A novel monarch butterfly optimization with global position updating operator for large-scale 0-1 knapsack problems,” Mathematics, vol. 7, no. 11, p. 1056, 2019.
[15] L. Sun, S. Chen, J. Xu, and Y. Tian, “Improved monarch butterfly optimization algorithm based on opposition-based learning and random local perturbation,” Complexity, vol. 2019, Article ID 4182148, 2019.
[16] V. Pasupuleti and C. Balaswamy, “Performance analysis of fractional earthworm optimization algorithm for optimal routing in wireless sensor networks,” ICST Transactions on Scalable Information Systems, vol. 8, no. 32, Article ID 169419, 2018.