mathematics

MDPI

Article

A Computational Method for Subdivision Depth of Ternary Schemes

Faheem Khan [1], Ghulam Mustafa [2], Aamir Shahzad [1], Dumitru Baleanu [3,4,5,*] and Maysaa M. Al-Qurashi [6]

1. Department of Mathematics, University of Sargodha, Sargodha 40100, Pakistan; fahimscholar@gmail.com (F.K.); uos.amir@gmail.com (A.S.)
2. Department of Mathematics, The Islamia University of Bahawalpur, Bahawalpur 63100, Pakistan; ghulam.mustafa@iub.edu.pk
3. Department of Mathematics, Cankaya University, Ankara 06530, Turkey
4. Institute of Space Sciences, 077125 Magurele-Bucharest, Romania
5. Department of Medical Research, China Medical University Hospital, China Medical University, Taichung 40402, Taiwan
6. Department of Mathematics, King Saud University, Riyadh 11495, Saudi Arabia; maysaa@ksu.edu.sa

* Correspondence: dumitru@cankaya.edu.tr or Baleanu@mail.cmuh.org.tw

Received: 16 March 2020; Accepted: 21 April 2020; Published: 18 May 2020

check for updates

Abstract: Subdivision schemes are extensively used in scientific and practical applications to produce continuous shapes in an iterative way. This paper introduces a framework to compute subdivision depths of ternary schemes. We first use subdivision algorithm in terms of convolution to compute the error bounds between two successive polygons produced by refinement procedure of subdivision schemes. Then, a formula for computing bound between the polygon at k-th stage and the limiting polygon is derived. After that, we predict numerically the number of subdivision steps (depths) required for smooth limiting shape based on the demand of user specified error (distance) tolerance. In addition, extensive numerical experiments were carried out to check the numerical outcomes of this new framework. The proposed methods are more efficient than the method proposed by Song et al.

Keywords: subdivision schemes; convolution; error bounds; subdivision depth; subdivision level

MSC: 65D17; 65D05; 65U07

1. Introduction

A broad and eminent area in Computer Aided Geometric Design (CAGD) deals with curves, surfaces, and their computational aspects. Subdivision is the most remarkable field for the purpose of modeling of curves and surfaces in CAGD. Subdivision methods have achieved much popularity in the past few years because of their implementation along with their mathematical formulation. The convolution technique [1] is one of the techniques used to merge different schemes. It has an important role in error analysis of the schemes. Actually, subdivision schemes take the polygons as input and successively produced smooth polygons or shapes as an output. Initially, the schemes with two rules were introduced. Later on, the interest was developed to recommend the schemes with three rules. This means at each subdivision level, every edge of the polygon is divided into three sub-edges. In the literature, these schemes are known as ternary schemes. Here, first we present brief review of these schemes then we will address the problem of error analysis.

Here we present an overview of the ternary subdivision schemes. Mustafa et al. [2–5] presented a $(2n-1)$-point ternary approximating and interpolating scheme, a 6-point ternary interpolating scheme, a family of even points ternary schemes, the odd point ternary approximating scheme respectively. Hassan et al. [6] and Siddiqi and Rehan [7] threw the light on 4-point ternary interpolating subdivision schemes. Kwan et al. [8] explored the phenomenon of 4-points ternary approximating scheme and Mustafa et al. [9,10] examined 5-point and 6-point ternary interpolating schemes and their differentiability. Siddiqi et al. [11,12] explained 4-point ternary interpolating scheme for curve sketching and constructed different ternary approximating subdivision schemes. Peng et al. [13,14] discovered non-linear circle preserving interpolating scheme and fractal behavior of ternary rational interpolating scheme. Further discussing on ternary subdivision scheme, Aslam [15] and Beccari et al. [16] showed their talent to highlight a family of 5-point non-linear ternary interpolating scheme and an interpolating 4-point ternary non-stationary scheme with tension control respectively. Certainly, there are a few methods for estimating the error bounds of these schemes.

Some of the authors [17–19] computed the error and order of the convergence of some binary schemes. Mustafa et al. [20–23] computed error bounds for binary, ternary, tensor product binary volumetric model and binary non-stationary schemes. Error bounds for a class of subdivision schemes based on the two-scale refinement equation were computed by Moncayo and Amat [24]. A formula for estimating the deviation of a binary interpolating subdivision curve from its data polygon was presented by Deng et al. [25]. However, the generalization of this formula to deal with the cases of n-ary interpolating and approximating schemes is still an open question. The following open question also arises in our mind: "How many subdivision steps (depths) are required to satisfy a user specified error (distance) tolerance?" Some of the researchers can be nominated as the embarking volunteers for the explanation of above questions such as: Mustafa and Hashmi [26] estimated subdivision depth computation for n-ary schemes by using first forward difference technique. Mustafa [27] presented subdivision depth computation technique for tensor product ternary volumetric model. Mustafa et al. also computed subdivision depth for triangular surfaces [28]. The above methods do not work for all type of subdivision schemes. Counter examples are also presented in this paper.

A novel numerical algorithm to estimate the subdivision depth was offered only for binary subdivision schemes in [29]. Still there is a gap/space to work for the subdivision depth of higher arity (i.e., ternary, quaternarys and so on) schemes. In this paper, an optimal approach is proposed to estimate subdivision depths for ternary (i.e., for each subdivision level, every edge of polygon is divided into three sub-edges) subdivision schemes.

The remaining part of the paper is arranged as follows. In Section 2, basic results, subdivision depths and numerical experiments of the method for univariate cases of the schemes are presented while in Section 3, these results for bivariate cases of the schemes are offered. Conclusions are drawn in Section 4.

2. Preliminary Results for Univariate Case

Let $\{p_i^k; i \in \mathbb{Z}\}$ be a sequence of 2D points in $\mathbf{R}^2$ which are obtained by the following refinement procedure

$$p_{3i+s}^{k+1} = \sum_{m=0}^{N-1} a_{s,m} p_{i+m}^{k}, \qquad s = 0, 1, 2, \tag{1}$$

with

$$\sum_{m=0}^{N-1} a_{s,m} = 1, \qquad s = 0, 1, 2, \tag{2}$$

where $k \geq 0$ indicates the refinement level. The points at 0th level p_i^0 are known as initial control points. The refinement procedure described in (1) along with its necessary condition of converge (2) is known as univariate ternary subdivision scheme. The following formulation of unknown coefficients $a_{s,m}$ is given by [21].

$$\begin{cases} b_{0,m} = \sum_{l=0}^{m} (a_{0,l} - a_{1,l}), \\ b_{1,m} = \sum_{l=0}^{m} (a_{1,l} - a_{2,l}), \\ b_{2,m} = a_{0,m} - (b_{0,m} + b_{1,m}), \end{cases}$$

with

$$\sum_{m=0}^{N-1} |b_{0,m}| < 1, \quad \sum_{m=0}^{N-1} |b_{1,m}| < 1, \quad \sum_{m=0}^{N-1} |b_{2,m}| < 1.$$

The following symbolization will also be used in coming section of this paper.

$$\begin{cases} d_{3m} = b_{0,m}, \\ d_{3m+1} = b_{1,m}, \\ d_{3m+2} = b_{2,m}. \end{cases} \tag{3}$$

Now we follow the techniques and notations presented in [24]. To be precise, let the vector $u_i = u_i^k$ represent the approximation coefficients associated with a certain k^{th} level of resolution. If $u_i^0 = u_i^{k;0}$ represents the k^{th} resolution level then the reconstruction algorithm used to define the approximation coefficients at stage $k+1$ in terms of the coefficient at stage k are obtained by the use of subdivision algorithms in terms of convolutions i.e.,

$$u_i^{k+1} = \sum_{n \in \mathbb{N}} d_{i-3n} u_n^k = (u^{k;0} \star d)_i, \tag{4}$$

where $\star$ denotes the convolution product of two vectors $u^{k;0}$ and $d = (d_n)_{n \in \mathbb{N}}$.

Generally, the convolution product of two vectors $u = (u_n)_{n \geq 0}$ and $v = (v_n)_{n \geq 0}$ of finite lengths l_u and l_v respectively for ternary subdivision scheme is defined as

$$(u \star v)_j = \sum_{n=max\{j-(l_v-1),0\}}^{min\{j,l_u-1\}} u_n v_{j-3n}, j = 0, 1, \ldots, l_u + l_v - 2. \tag{5}$$

In the following subsection, we present the generalized version of the results presented in the Appendices A1 and A2 of [24].

2.1. Reformulation of Successive Convolutions

In this subsection, we obtain some generalized inequalities used in order to find the subdivision depth of ternary subdivision schemes for the generation of curves. Their further generalizations are presented in Section 3 for the computation of depth of ternary subdivision schemes for tensor product surfaces. This section contains typical rigorous and tedious mathematical expressions. Readers are refers to Example 1 of this section for better understanding.

Lemma 1. *Let $u = u_n$ be the vector of finite length and $d = \{d_n\}_{n=0}^{3N-1}$ with $d_n = 0$ for $n \geq 3N$, then the following one dimensional k_0 convolutions is bounded by*

$$\|((\ldots(((u^{(0)} \star d)^{(0)}) \star d)^{(0)} \star \ldots \star d)^{(0)} \star d)\|_\infty \leq \|u\|_\infty \sup_j \left\{ \sum_{m=0}^{\lfloor j/3^{k_0} \rfloor} |B_{m,j}^{k_0}| \right\},$$

where $B_{m,j}^{k_0}$ is defined recursively by

$$\begin{cases} B_{m,j}^1 = d_{j-3m}, \\ B_{m,j}^{k_0} = \sum_{p=3m}^{\lfloor j/3^{k_0-1} \rfloor} B_{m,p}^1 B_{p,j}^{k_0-1}, \qquad k_0 \geq 2, \end{cases}$$

and

$$j \in \Sigma(k_0, N) = \{\Omega(k_0, N) - 3^{k_0} + 1, \Omega(k_0, N) - 3^{k_0} + 2, \ldots, \Omega(k_0, N)\},$$
$$\Omega(k_0, N) = (3^{k_0} - 2)(3N - 1).$$

Proof. To prove this result, we start with the case of $k_0 = 1$ and $k_0 = 2$ convolutions and then a general case will be derived.

Case $k_0 = 1$: From (5), we obtain a relation given in the following

$$|(u^{(0)} \star d)_j| = \left| \sum_{n=0}^{\lfloor j/3 \rfloor} u_n d_{j-3n} \right|, \tag{6}$$

where $\lfloor . \rfloor$ denotes the integer part. Using infinity norm $\left(\|u\|_\infty = \max\{|u_o|, \ldots, |u_{\lfloor j/3 \rfloor}|\}\right)$, we get

$$|(u^{(0)} \star d)_j| \quad \leq \quad \|u\|_\infty \sum_{n=0}^{\lfloor j/3 \rfloor} |d_{j-3n}|.$$

Now

$$\sup |(u^{(0)} \star d)_j| \quad \leq \quad \sup \left(\|u\|_\infty \sum_{n=0}^{\lfloor j/3 \rfloor} |d_{j-3n}| \right).$$

This infers

$$\sup |(u^{(0)} \star d)_j| \quad \leq \quad \|u\|_\infty \sup \left(\sum_{n=0}^{\lfloor j/3 \rfloor} |B_{n,j}^1| \right),$$

where

$$d_{j-3n} = B_{n,j}^1. \tag{7}$$

Thus,

$$\|(u^{(0)} \star d)_j\|_\infty \leq \|u\|_\infty \sup \left(\sum_{n=0}^{\lfloor j/3 \rfloor} |B_{n,j}^1| \right).$$

Case $k_0 = 2$: From (6), we acquire

$$((u^{(0)} \star d)^{(0)} \star d)_j = \sum_{m=0}^{\lfloor j/3 \rfloor} (u^{(0)} \star d)_m d_{j-3m} = \sum_{m=0}^{\lfloor j/3 \rfloor} \left(\sum_{n=0}^{\lfloor m/3 \rfloor} u_n d_{m-3n} \right) d_{j-3m}.$$

This infers

$$\begin{aligned} ((u^{(0)} \star d)^{(0)} \star d)_j &= u_0 d_0 d_j + u_0 d_1 d_{j-3} + u_0 d_2 d_{j-6} + u_0 d_3 d_{j-9} + u_1 d_0 d_{j-9} + u_0 d_4 d_{j-12} \\ &+ u_1 d_1 d_{j-12} + \ldots + u_0 d_{\lfloor \frac{j}{3} \rfloor} d_0 + u_1 d_{\lfloor \frac{j}{3} \rfloor - 3} d_0 + \ldots + u_{\lfloor \frac{j}{3^2} \rfloor} d_{\lfloor \frac{j}{3} \rfloor - 3 \lfloor \frac{j}{3^2} \rfloor} d_0. \end{aligned}$$

This implies that

$$((u^{(0)} \star d)^{(0)} \star d)_j = \sum_{m=0}^{\lfloor j/3^2 \rfloor} u_m \left(\sum_{n=3m}^{\lfloor j/3 \rfloor} d_{n-3m} d_{j-3n} \right).$$

This infers

$$((u^{(0)} \star d)^{(0)} \star d)_j = \sum_{m=0}^{\lfloor j/3^2 \rfloor} u_m \left(\sum_{n=3m}^{\lfloor j/3 \rfloor} B^1_{m,n} B^1_{n,j} \right) = \sum_{m=0}^{\lfloor j/3^2 \rfloor} u_m B^2_{m,j},$$

where

$$B^2_{m,j} = \sum_{n=3m}^{\lfloor j/3 \rfloor} B^1_{m,n} B^1_{n,j}. \tag{8}$$

So

$$|((u^{(0)} \star d)^{(0)} \star d)_j| = \left| \sum_{m=0}^{\lfloor j/3^2 \rfloor} u_m B^2_{m,j} \right| \leq \|u\|_\infty \sum_{m=0}^{\lfloor j/3^2 \rfloor} |B^2_{m,j}|.$$

This implies

$$\|((u^{(0)} \star d)^{(0)} \star d)_j\|_\infty \leq \|u\|_\infty \sup_j \left(\sum_{m=0}^{\lfloor j/3^2 \rfloor} |B^2_{m,j}| \right).$$

General case: By using the same technique, we acquire the reformulations for k_0-th convolutions, which is in the following

$$((\ldots(((u^{(0)} \star d)^{(0)}) \star d)^{(0)} \star \ldots \star d)^{(0)} \star d)_j = \sum_{m=0}^{\lfloor j/3^{k_0} \rfloor} u_m B^{k_0}_{m,j}.$$

Which implies

$$\|((\ldots(((u^{(0)} \star d)^{(0)}) \star d)^{(0)} \star \ldots \star d)^{(0)} \star d)\|_\infty \leq \|u\|_\infty \sup_j \left\{ \sum_{m=0}^{\lfloor j/3^{k_0} \rfloor} |B^{k_0}_{m,j}| \right\}. \tag{9}$$

□

Lemma 2. *The term $B^{k_0}_{m,j}$ in the inequality (9) has the following expression*

$$B^{k_0}_{m-1,j-3^{k_0}} = B^{k_0}_{m,j} = B^{k_0}_{m+1,j+3^{k_0}}. \tag{10}$$

Proof. Now we start for an induction process, which is over k_0. Then
Case $k_0 = 1$**:**

$$B^1_{m,j} = d_{j-3m} = d_{j+3-3(m+1)} = B^1_{m+1,j+3}. \tag{11}$$

Similarly

$$B^1_{m+1,j} = d_{j-3(m+1)} = B^1_{m,j-3}. \tag{12}$$

From (8), we have

$$B^2_{m,j} = \sum_{n=3m}^{\lfloor j/3 \rfloor} B^1_{m,n} B^1_{n,j}.$$

Using (11), we have

$$B^2_{m,j} = \sum_{n=3m}^{\lfloor j/3 \rfloor} B^1_{m,n} B^1_{n+1,j+3}. \tag{13}$$

Now replace n by $n-3$ in (13), we obtain

$$B^2_{m,j} = \sum_{n=3(m-1)}^{\lfloor j/3+3 \rfloor} B^1_{m,n-3} B^1_{n-2,j+3}.$$

Now using (12), we acquire

$$B^2_{m,j} = \sum_{n=3(m-1)}^{\lfloor j/3+3 \rfloor} B^1_{m+1,n} B^1_{n,j+3^2}.$$

so

$$B^2_{m,j} = B^2_{m+1,j+3^2}.$$

We suppose that it is true for an integer $k_0 = M$, that is

$$B^M_{m,j} = B^M_{m+1,j+3^M}. \tag{14}$$

Case $k_0 = M+1$**:** Consider

$$B^{M+1}_{m,j} = \sum_{n=3m}^{\lfloor j/3^M \rfloor} B^1_{m,n} B^M_{n,j}.$$

Using (14), we have

$$B^{M+1}_{m,j} = \sum_{n=3m}^{\lfloor j/3^M \rfloor} B^1_{m,n} B^M_{n+1,j+3^M}. \tag{15}$$

Now, replace n by $n-3$ in (15), we have

$$B^{M+1}_{m,j} = \sum_{n=3(m+1)}^{\lfloor j/3^M+3 \rfloor} B^1_{m,n-3} B^M_{n-2,j+3^M}.$$

Using (12) and (14), we acquire

$$B^{M+1}_{m,j} = B^{M+1}_{m+1,j+3^{M+1}}.$$

Similarly we can prove

$$B^{M+1}_{m,j} = B^{M+1}_{m-1,j-3^{M+1}}.$$

Hence

$$B^{k_0}_{m-1,j-3^{k_0}} = B^{k_0}_{m,j} = B^{k_0}_{m+1,j+3^{k_0}}.$$

□

Now, applying Lemmas 1 and 2, we arrive at the following useful result:

Corollary 1. *The associated constant of a k_0-th convolution with vector $d = \{d_0, d_1, \ldots, d_{3N-1}\}$ is*

$$D_{k_0} = \sup_j \left\{ \sum_{m=0}^{\lfloor j/3^{k_0} \rfloor} |B^{k_0}_{m,j}| \right\} = \sup_{j\in\Sigma(k_0,N)} \left\{ \sum_{m=0}^{\lfloor j/3^{k_0} \rfloor} |B^{k_0}_{m,j}| \right\}. \tag{16}$$

Proof. Assume that $d = \{d_0, d_1, \ldots, d_{3N-1}\}$, with $N \in \mathbb{N}$ and $\Omega(k_0, N) = (3^{k_0} - 2)(3N - 1)$. Then for $j > \Omega(k_0, N)$ and by using Lemma 1, we acquire

$$B^{k_0}_{0,j} = 0. \tag{17}$$

Similarly for $j > \Omega(k_0, N) + m3^{k_0}$ and using Lemma 2, we have

$$B^{k_0}_{m,j} = 0. \tag{18}$$

Finally, using (17) and (18), we get (16). □

2.2. Subdivision Depth for Ternary Subdivision Curves

In this section, we first generalize the inequalities (2.18) and then (2.5) which were presented in [21]. After that, we present a numerical inequality to compute the subdivision depth of ternary subdivision schemes for curve modeling.

Theorem 1. *Consider the initial polygon $p^0_i, i \in \mathbb{Z}$ and $p^k_i, k \geq 0$, recursively interpreted by (1) together with (2). Suppose P^k represents the polygon at the points $\{p^k_i\}$. Then after two successive refinements/iterations k and $k+1$, the error bounds between these two iterations is*

$$\|P^{k+1} - P^k\|_\infty \leq \xi\eta(D_{k_0})^k,$$

where $D_{k_0}, k_0 \geq 1$ defined in (16), $\eta = \max_i \|p^0_{i+1} - p^0_i\|$ and

$$\begin{aligned} \xi &= \max\left(\sum_{m=0}^{N-2} |\tilde{a}_{0,m}|, \sum_{m=0}^{N-2} |\tilde{a}_{1,m}|, \sum_{m=0}^{N-2} |\tilde{a}_{2,m}| \right), \quad \tilde{a}_{s,m} = \sum_{i=m+1}^{N-1} a_{s,i}, 0 \leq s \leq 2, \\ \tilde{a}_{1,0} &= \sum_{i=1}^{N-1} a_{1,i} - \frac{1}{3}, \quad \tilde{a}_{2,0} = \sum_{i=1}^{N-1} a_{2,i} - \frac{2}{3}. \end{aligned}$$

Proof. See in [21]. □

Theorem 2. *Let a limit curve P^∞ be linked with the subdivision iterative process, then under the same conditions used in Theorem 1 the following inequality hold*

$$\left\|P^\infty - P^k\right\|_\infty \leq \xi\eta\left(\frac{(D_{k_0})^k}{1-D_{k_0}}\right),$$

where $k_0 \geq 1$ is a natural number, such that $D_{k_0} < 1$.

Proof. See in [21]. □

Theorem 3. *Let k be subdivision depth and let ∇^k be the error bound between ternary subdivision curve P^∞ and its k-level control polygon P^k. For arbitrary $\epsilon > 0$, if*

$$k \geq log_{D_{k_0}}\left(\frac{\epsilon(1-D_{k_0})}{\xi\eta}\right), \tag{19}$$

then $\nabla^k \leq \epsilon$.

Proof. Let ∇^k be the distance between limit curve P^∞ and control polygon P^k at k-th level defined in Theorem 2, such that

$$\nabla^k = \left\|P^\infty - P^k\right\|_\infty \leq \xi\eta\left(\frac{(D_{k_0})^k}{1-D_{k_0}}\right).$$

To obtain given error tolerance $\epsilon > 0$, consider

$$\xi\eta\left(\frac{(D_{k_0})^k}{1-D_{k_0}}\right) \leq \epsilon,$$

which implies

$$\frac{\xi\eta}{\epsilon(1-D_{k_0})} \leq (D_{k_0}^{-1})^k.$$

Now taking logarithm, we have

$$k \quad \geq \quad \frac{\log\left(\frac{\xi\eta}{\epsilon(1-D_{k_0})}\right)}{\log D_{k_0}^{-1}} = \frac{\log\left(\frac{\xi\eta}{\epsilon(1-D_{k_0})}\right)}{-\log D_{k_0}} = -\log_{D_{k_0}}\left(\frac{\xi\eta}{\epsilon(1-D_{k_0})}\right) = \log_{D_{k_0}}\left(\frac{\xi\eta}{\epsilon(1-D_{k_0})}\right)^{-1},$$

which implies

$$k \geq \log_{D_{k_0}}\left(\frac{\epsilon(1-D_{k_0})}{\xi\eta}\right),$$

then $\nabla^k \leq \epsilon$. This completes the proof. □

2.3. Application for Univariate Case

Here, we present a few numerical experiments to compute subdivision depths of ternary subdivision schemes for curves. The associated constants $D_{k_0}, k_0 \geq 1$ defined in (16) of some ternary subdivision curves are shown in Table 1.

Table 1. Associated constants of ternary subdivision curves.

Scheme/ D_{k_0}	$D_1=\delta$	D_2	D_3	D_4	D_5
2-point scheme [12]	0.333333	0.111111	0.037037	0.012346	0.004115
3-point scheme [30]	0.555467	0.253017	0.11107	0.047856	0.020353
4-point scheme [8]	0.441358	0.176393	0.067478	0.027136	0.010728
4-point scheme [6]	0.444444	0.171682	0.070918	0.029278	0.012089

Remark 1. *In this technique D_{k_0} for $k_0 = 1$ is equal to δ defined in [21]. Please note that in [21], if $\delta > 1$ then error bounds cannot be computed. However, in the proposed technique, if we increase the value of k_0 until D_{k_0} becomes less than one, so using this argument we can compute error bounds in each situation even though the value of δ becomes greater or equal to one.*

Example 1. *Given initial polygon $p_i^0 = p_i, i \in \mathbb{Z}$ with values $p_i^k, k \geq 1$ be interpreted recursively by the 2-point ternary approximating subdivision scheme [12] (i.e., $a_{0,0} = \frac{5}{6}$, $a_{0,1} = \frac{1}{6}$, $a_{1,0} = \frac{1}{2}$, $a_{1,1} = \frac{1}{2}$, $a_{2,0} = \frac{1}{6}$, $a_{2,1} = \frac{5}{6}$). For this ternary two point scheme ($N = 2$), we have from (16)*

$$D_{k_0} = \sup_{j\in\Sigma(k_0,2)} \left\{ \sum_{m=0}^{\lfloor j/3^{k_0} \rfloor} |B_{m,j}^{k_0}| \right\}.$$

For $k_0 = 1$, we get

$$D_1 = \sup_{j\in\Sigma(1,2)} \left\{ \sum_{m=0}^{\lfloor j/3 \rfloor} |B_{m,j}^{1}| \right\} = \sup_{j\in\{3,4,5\}} \left\{ \sum_{m=0}^{\lfloor j/3 \rfloor} |d_{j-3m}| \right\}.$$

Using (3) and Lemma 1, we have $d = \{d_n\}_{n=0}^{5}$ with $d_n = 0$ for $n \geq 6$. Hence

$$\{d_0, d_1, d_2, d_3, d_4, d_5\} = \left\{ \frac{1}{3}, \frac{1}{3}, \frac{1}{6}, 0, 0, \frac{1}{6} \right\}.$$

Now consider

$$D_1 = \sup \left\{ \sum_{m=0}^{\lfloor 3/3 \rfloor} |d_{3-3m}|, \sum_{m=0}^{\lfloor 4/3 \rfloor} |d_{4-3m}|, \sum_{m=0}^{\lfloor 5/3 \rfloor} |d_{5-3m}| \right\}.$$

This implies

$$\begin{aligned} D_1 &= \sup \Big\{ |d_3| + |d_0|, |d_4| + |d_1|, |d_5| + |d_2| \Big\} \\ &= \sup \left\{ |0| + \left|\frac{1}{3}\right|, |0| + \left|\frac{1}{3}\right|, \left|\frac{1}{6}\right| + \left|\frac{1}{6}\right| \right\} = \frac{1}{3}. \end{aligned}$$

For $k_0 = 2$, we get

$$\begin{aligned} D_2 &= \sup_{j\in\Sigma(2,2)} \left\{ \sum_{m=0}^{\lfloor j/3^2 \rfloor} |B_{m,j}^{2}| \right\} = \sup_{j\in\{27,28,\ldots,35\}} \left\{ \sum_{m=0}^{\lfloor j/9 \rfloor} |B_{m,j}^{2}| \right\} \\ &= \sup_{j\in\{27,28,\ldots,35\}} \left\{ \sum_{m=0}^{\lfloor j/9 \rfloor} \left| \sum_{n=3m}^{\lfloor j/3 \rfloor} B_{m,n}^{1} B_{n,j}^{1} \right| \right\}. \end{aligned}$$

This implies

$$
\begin{aligned}
D_2 \;=\; \sup\Bigg\{ & \sum_{m=0}^{\lfloor 27/9\rfloor}\left|\sum_{n=3m}^{\lfloor 27/3\rfloor} B^1_{m,n}B^1_{n,27}\right|, \sum_{m=0}^{\lfloor 28/9\rfloor}\left|\sum_{n=3m}^{\lfloor 28/3\rfloor} B^1_{m,n}B^1_{n,28}\right|, \sum_{m=0}^{\lfloor 29/9\rfloor}\left|\sum_{n=3m}^{\lfloor 29/3\rfloor} B^1_{m,n}B^1_{n,29}\right|, \\
& \sum_{m=0}^{\lfloor 30/9\rfloor}\left|\sum_{n=3m}^{\lfloor 30/3\rfloor} B^1_{m,n}B^1_{n,30}\right|, \sum_{m=0}^{\lfloor 31/9\rfloor}\left|\sum_{n=3m}^{\lfloor 31/3\rfloor} B^1_{m,n}B^1_{n,31}\right|, \sum_{m=0}^{\lfloor 32/9\rfloor}\left|\sum_{n=3m}^{\lfloor 32/3\rfloor} B^1_{m,n}B^1_{n,32}\right|, \\
& \sum_{m=0}^{\lfloor 33/9\rfloor}\left|\sum_{n=3m}^{\lfloor 33/3\rfloor} B^1_{m,n}B^1_{n,33}\right|, \sum_{m=0}^{\lfloor 34/9\rfloor}\left|\sum_{n=3m}^{\lfloor 34/3\rfloor} B^1_{m,n}B^1_{n,34}\right|, \sum_{m=0}^{\lfloor 35/9\rfloor}\left|\sum_{n=3m}^{\lfloor 35/3\rfloor} B^1_{m,n}B^1_{n,35}\right|\Bigg\}.
\end{aligned}
$$

This further implies

$$
D_2 = \sup\Big\{\chi_1, \chi_2, \chi_3, \chi_4, \chi_5, \chi_6, \chi_7, \chi_8, \chi_9\Big\},
$$

where

$$
\begin{aligned}
\chi_1 = & \Big|d_0d_{27}+d_1d_{24}+d_2d_{21}+d_3d_{18}+d_4d_{15}+d_5d_{12}+d_6d_9+d_7d_6+d_8d_3+d_9d_0\Big| + \Big|d_0d_{18} \\
& +d_1d_{15}+d_2d_{12}+d_3d_9+d_4d_6+d_5d_3+d_6d_0\Big| + \Big|d_0d_9+d_1d_6+d_2d_3+d_3d_0\Big| + \Big|d_0d_0\Big|,
\end{aligned}
$$

$$
\begin{aligned}
\chi_2 = & \Big|d_0d_{28}+d_1d_{25}+d_2d_{22}+d_3d_{19}+d_4d_{16}+d_5d_{13}+d_6d_{10}+d_7d_7+d_8d_4+d_9d_1\Big| + \Big|d_0d_{19} \\
& +d_1d_{16}+d_2d_{13}+d_3d_{10}+d_4d_7+d_5d_4+d_6d_1\Big| + \Big|d_0d_{10}+d_1d_7+d_2d_4+d_3d_1\Big| + \Big|d_0d_1\Big|,
\end{aligned}
$$

$$
\begin{aligned}
\chi_3 = & \Big|d_0d_{29}+d_1d_{26}+d_2d_{23}+d_3d_{20}+d_4d_{17}+d_5d_{14}+d_6d_{11}+d_7d_8+d_8d_5+d_9d_2\Big| + \Big|d_0d_{20} \\
& +d_1d_{17}+d_2d_{14}+d_3d_{11}+d_4d_8+d_5d_5+d_6d_2\Big| + \Big|d_0d_{11}+d_1d_8+d_2d_5+d_3d_2\Big| + \Big|d_0d_2\Big|,
\end{aligned}
$$

$$
\begin{aligned}
\chi_4 = & \Big|d_0d_{30}+d_1d_{27}+d_2d_{24}+d_3d_{21}+d_4d_{18}+d_5d_{15}+d_6d_{12}+d_7d_9+d_8d_6+d_9d_3+d_{10}d_0\Big| \\
& +\Big|d_0d_{21}+d_1d_{18}+d_2d_{15}+d_3d_{12}+d_4d_9+d_5d_6+d_6d_3+d_7d_0\Big| + \Big|d_0d_{12}+d_1d_9+d_2d_6 \\
& +d_3d_3+d_4d_0\Big| + \Big|d_0d_3+d_1d_0\Big|,
\end{aligned}
$$

$$\begin{aligned}
\chi_5 &= \Big|d_0d_{31} + d_1d_{28} + d_2d_{25} + d_3d_{22} + d_4d_{19} + d_5d_{16} + d_6d_{13} + d_7d_{10} + d_8d_7 + d_9d_4 + d_{10}d_1\Big| \\
&\quad + \Big|d_0d_{22} + d_1d_{19} + d_2d_{16} + d_3d_{13} + d_4d_{10} + d_5d_7 + d_6d_4 + d_7d_1\Big| + \Big|d_0d_{13} + d_1d_{10} + d_2d_7 \\
&\quad + d_3d_4 + d_4d_1\Big| + \Big|d_0d_4 + d_1d_1\Big|,
\end{aligned}$$

$$\begin{aligned}
\chi_6 &= \Big|d_0d_{32} + d_1d_{29} + d_2d_{26} + d_3d_{23} + d_4d_{20} + d_5d_{17} + d_6d_{14} + d_7d_{11} + d_8d_8 + d_9d_5 + d_{10}d_2\Big| \\
&\quad + \Big|d_0d_{23} + d_1d_{20} + d_2d_{17} + d_3d_{14} + d_4d_{11} + d_5d_8 + d_6d_5 + d_7d_2\Big| + \Big|d_0d_{14} + d_1d_{11} + d_2d_8 \\
&\quad + d_3d_5 + d_4d_2\Big| + \Big|d_0d_5 + d_1d_2\Big|,
\end{aligned}$$

$$\begin{aligned}
\chi_7 &= \Big|d_0d_{33} + d_1d_{30} + d_2d_{27} + d_3d_{24} + d_4d_{21} + d_5d_{18} + d_6d_{15} + d_7d_{12} + d_8d_9 + d_9d_6 + d_{10}d_3 \\
&\quad + d_{11}d_0\Big| + \Big|d_0d_{24} + d_1d_{21} + d_2d_{18} + d_3d_{15} + d_4d_{12} + d_5d_9 + d_6d_6 + d_7d_3 + d_8d_0\Big| + \Big|d_0d_{15} \\
&\quad + d_1d_{12} + d_2d_9 + d_3d_6 + d_4d_3 + d_5d_0\Big| + \Big|d_0d_6 + d_1d_3 + d_2d_0\Big|,
\end{aligned}$$

$$\begin{aligned}
\chi_8 &= \Big|d_0d_{34} + d_1d_{31} + d_2d_{28} + d_3d_{25} + d_4d_{22} + d_5d_{19} + d_6d_{16} + d_7d_{13} + d_8d_{10} + d_9d_7 + d_{10}d_4 \\
&\quad + d_{11}d_1\Big| + \Big|d_0d_{25} + d_1d_{22} + d_2d_{19} + d_3d_{16} + d_4d_{13} + d_5d_{10} + d_6d_7 + d_7d_4 + d_8d_1\Big| + \Big|d_0d_{16} \\
&\quad + d_1d_{13} + d_2d_{10} + d_3d_7 + d_4d_4 + d_5d_1\Big| + \Big|d_0d_7 + d_1d_4 + d_2d_1\Big|,
\end{aligned}$$

$$\begin{aligned}
\chi_9 &= \Big|d_0d_{35} + d_1d_{32} + d_2d_{29} + d_3d_{26} + d_4d_{23} + d_5d_{20} + d_6d_{17} + d_7d_{14} + d_8d_{11} + d_9d_8 + d_{10}d_5 \\
&\quad + d_{11}d_2\Big| + \Big|d_0d_{26} + d_1d_{23} + d_2d_{20} + d_3d_{17} + d_4d_{14} + d_5d_{11} + d_6d_8 + d_7d_5 + d_8d_2\Big| + \Big|d_0d_{17} \\
&\quad + d_1d_{14} + d_2d_{11} + d_3d_8 + d_4d_5 + d_5d_2\Big| + \Big|d_0d_8 + d_1d_5 + d_2d_2\Big|\Big\}.
\end{aligned}$$

Since $d_i = 0$, for all $i > 5$, so

$$\begin{aligned}
D_2 &= \sup\Big\{0 + |d_5d_3| + |d_2d_3 + d_3d_0| + |d_0d_0|, 0 + |d_5d_4| + |d_2d_4 + d_3d_1| + |d_0d_1|, 0 + |d_5d_5| \\
&\quad + |d_2d_5 + d_3d_2| + |d_0d_2|, 0 + 0 + |d_3d_3 + d_4d_0| + |d_0d_3 + d_1d_0|, 0 + 0 + |d_3d_4 + d_4d_1| \\
&\quad + |d_0d_4 + d_1d_1|, 0 + 0 + |d_3d_5 + d_4d_2| + |d_0d_5 + d_1d_2|, 0 + 0 + |d_4d_3 + d_5d_0| + |d_1d_3 + d_2d_0|, \\
&\quad 0 + 0 + |d_4d_4 + d_5d_1| + |d_1d_4 + d_2d_1|, 0 + 0 + |d_4d_5 + d_5d_2| + |d_1d_5 + d_2d_2|\Big\}.
\end{aligned}$$

This further implies

$$D_2 = \sup\left\{\frac{1}{9},\frac{1}{9},\frac{1}{9},\frac{1}{9},\frac{1}{9},\frac{1}{9},\frac{1}{9},\frac{1}{9},\frac{1}{9}\right\} = \frac{1}{9}.$$

Similarly, we can compute the values of $D_{k_0}, k_0 \geq 3$. For convenience, we have computed the values up to $k_0 = 5$, which are shown in Table 1. Its subdivision depth k (level of iterations) is computed by using Theorem 3 at different values of $D_{k_0}, k_0 \geq 1$ which are given in Table 2.

Table 2. Subdivision depth of 2-point approximating ternary subdivision curves.

D_{k_0}/ϵ	6.88×10^{-5}	2.83×10^{-7}	1.16×10^{-9}	4.79×10^{-12}	1.97×10^{-14}	8.12×10^{-17}	3.34×10^{-19}
$D_1 = \delta$	5	10	15	20	25	30	35
D_2	3	5	8	10	13	15	18
D_3	2	3	5	7	8	10	12
D_4	1	2	4	5	6	7	9
D_5	1	2	3	4	5	6	7

From this table, we observed that as k_0 increases subdivision depth decreases. This shows that the less subdivision depth can be obtained by using proposed technique. In other words, we need fewer iteration to get optimal subdivision depth as compared to the technique given in [21] (which is denoted by δ). For example, by [21], it needs thirty five iterations to obtain error tolerance $\epsilon = 3.34 \times 10^{-19}$ but using our technique, it needs only seven iterations corresponding to D_5. The comparison of first and fifth convolution results is shown in Figure 1a.

Example 2. *Consider the 3-point interpolating subdivision scheme [30] with $b = 0.2778, a = b - \frac{1}{3}$. Its subdivision depths k for $D_{k_0}, k_0 \geq 1$ (see, Table 1) are computed by using Theorem 3, which are shown in Table 3 and in graphical sense shown in Figure 1b.*

Table 3. Subdivision depth of 3-point interpolating ternary subdivision curves.

D_{k_0}/ϵ	1.38×10^{-3}	2.81×10^{-5}	5.73×10^{-7}	1.16×10^{-8}	2.37×10^{-10}	4.83×10^{-12}	9.84×10^{-14}
$D_1 = \delta$	8	15	21	28	34	41	48
D_2	3	6	9	12	14	17	20
D_3	2	4	5	7	9	11	12
D_4	1	3	4	5	6	8	9
D_5	1	2	3	4	5	6	7

Example 3. *Given initial control polygon $p_i^0 = p_i, i \in \mathbb{Z}$ with values $p_i^k, k \geq 1$ be illustrated recursively by the ternary 4-point approximating scheme [8].*

Its subdivision depth k for $D_{k_0}, k_0 \geq 1$ are given in Table 4. It is also demonstrated with the help of Figure 1c.

Table 4. Subdivision depth of 4-point approximating ternary subdivision curves.

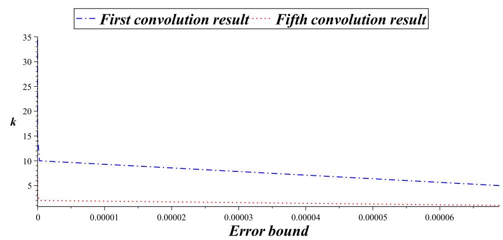
D_{k_0}/ϵ	1.26×10^{-3}	1.35×10^{-5}	1.45×10^{-7}	1.56×10^{-9}	1.67×10^{-11}	1.79×10^{-13}	1.92×10^{-15}
$D_1 = \delta$	6	12	17	23	28	34	40
D_2	3	5	8	11	13	16	18
D_3	2	3	5	7	8	10	12
D_4	1	3	4	5	6	8	9
D_5	1	2	3	4	5	6	7

Example 4. *Given $p_i^0 = p_i, i \in \mathbb{Z}$ be the initial polygon and for all positive integers we have the values p_i^k be specified recursively by ternary 4-point interpolating subdivision scheme [6] with parameter $w = \frac{1}{12}$.*

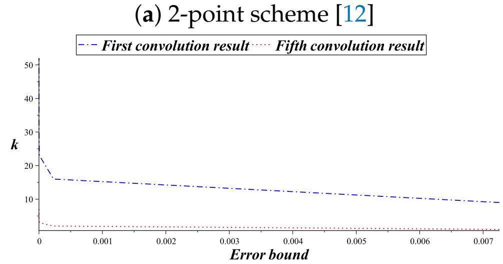
Its subdivision depths for $D_{k_0}, k_0 \geq 1$ (see, Table 1) are given in Table 5 and its performance is shown in Figure 1d.

Table 5. Subdivision depth of 4-point interpolating ternary subdivision curves.

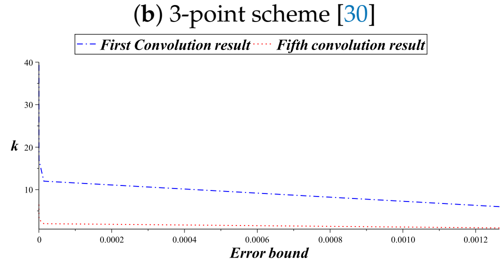
D_{k_0}/ϵ	1.22×10^{-3}	1.47×10^{-5}	1.78×10^{-7}	2.16×10^{-9}	2.61×10^{-11}	3.15×10^{-13}	3.81×10^{-15}
$D_1 = \delta$	6	12	17	22	28	33	39
D_2	3	5	8	10	13	15	18
D_3	2	3	5	7	8	10	12
D_4	1	2	4	5	6	7	9
D_5	1	2	3	4	5	6	7



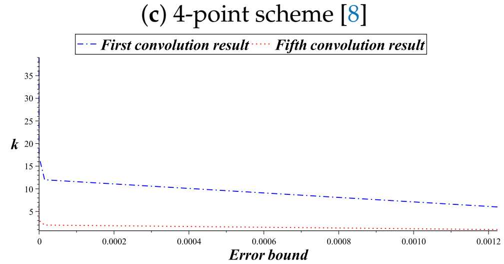
(**a**) 2-point scheme [12]



(**b**) 3-point scheme [30]



(**c**) 4-point scheme [8]



(**d**) 4-point scheme [6]

Figure 1. Comparison between first and fifth convolutions. This shows that the error decreases with the increase of convolution. Of course, it decreases with the increase of subdivision depth.

3. Preliminary Results for Bivariate Case

In this section, we generalize our representation of the 2-dimensional case to the 3-dimensional case. That is, we first focus our attention on generalizing the inequalities presented in Section 2.1 then we generalize the inequalities of Section 2.2 to compute subdivision depth of tensor product surfaces. For this, let $\{p^k_{i,j}; i,j \in \mathbb{Z}\}$ be the sequence of 3D points $\mathbf{R^N}$, $\mathbf{N} \geq 3$ which are produced by the following tensor product of ternary scheme (1)

$$p^{k+1}_{3i+\alpha,3j+\beta} = \sum_{r=0}^{N-1}\sum_{s=0}^{N-1} a_{\alpha,r}a_{\beta,s}p^k_{i+r,m+s}, \alpha,\beta = 0,1,2, \tag{20}$$

where $a_{\alpha,r}$ satisfies (2).

Now we assign the coefficients $f = \{f_n\}_{n\in\mathbb{N}}$ and $g = \{g_n\}_{n\in\mathbb{N}}$, by using the same procedure of symbolization given in [21] i.e.,

$$\begin{cases} f_{3r} = a_{0,N-r-1}, \\ f_{3r+1} = a_{1,N-r-1}, \\ f_{3r+2} = a_{2,N-r-1} & r = 0,...,N-1. \\ g_{3s} = b_{0,N-s-1} \\ g_{3s+1} = b_{1,N-s-1} \\ g_{3s+2} = b_{2,N-s-1} & s = 0,...,N-1. \end{cases}$$

To achieve the goal, all that is needed is to make the set up given before Section 2.1 for the 3D case. Here we skip the unnecessary detail and directly go to the following results.

Lemma 3. *Let $u = u_{m,n}$ be the vector of finite length for bivariate case and $f = \{f_n\}_{n=0}^{3N-1}$, $g = \{g_n\}_{n=0}^{3N-1}$ with $f_n = g_n = 0$ for $n \geq 3N$, then the following two dimensional k_0 convolutions are bounded by*

$$\max_{i,j} |u^{k_0}_{i,j}| \leq F_{k_0}G_{k_0} \max_{m,n} |u^0_{m,n}|,$$

where

$$F_{k_0} = \max_i \left\{ \sum_{m=0}^{\lfloor i/3^{k_0} \rfloor} |B^{k_0,f}_{m,i}| \right\}$$

and

$$G_{k_0} = \max_j \left\{ \sum_{n=0}^{\lfloor j/3^{k_0} \rfloor} |B^{k_0,g}_{n,j}| \right\}.$$

Proof. To prove the result, we start with the case of $k_0 = 1$ and $k_0 = 2$ convolutions and later on we analyze the general case.
Case $k_0 = 1$: Consider an arbitrary sequence of vectors $u_{i,j}$. Then we have

$$u^{k_0}_{i,j} = (u^{k_0-1;0} \star fg)_{i,j} = \sum_{m=0}^{\lfloor i/3 \rfloor}\sum_{n=0}^{\lfloor j/3 \rfloor} u^{k_0-1}_{m,n} f_{i-3m}g_{j-3n},$$

where we are taking $B^{1,f}_{m,i} = f_{i-3m}$ and $g_{j-3n} = B^{1,g}_{n,j}$ for arbitrary sequence f and g. Thus,

$$u^{k_0}_{i,j} = (u^{k_0-1;0} \star fg)_{i,j} = \sum_{m=0}^{\lfloor i/3 \rfloor} \sum_{n=0}^{\lfloor j/3 \rfloor} u^{k_0-1}_{m,n} B^{1,f}_{m,i} B^{1,g}_{n,j}.$$

This implies

$$\begin{aligned} \max_{i,j} |u^{k_0}_{i,j}| &= \max_{i,j} \left| \sum_{m=0}^{\lfloor i/3 \rfloor} \sum_{n=0}^{\lfloor j/3 \rfloor} u^{k_0-1}_{m,n} B^{1,f}_{m,i} B^{1,g}_{n,j} \right|, \\ &\leq \max_{i,j} \sum_{m=0}^{\lfloor i/3 \rfloor} \sum_{n=0}^{\lfloor j/3 \rfloor} |B^{1,f}_{m,i}||B^{1,g}_{n,j}| \max_{m,n} |u^{k_0-1}_{m,n}|. \end{aligned} \tag{21}$$

Consider

$$F_1 = \max_i \left\{ \sum_{m=0}^{\lfloor i/3 \rfloor} |B^{1,f}_{m,i}| \right\}$$

and

$$G_1 = \max_j \left\{ \sum_{n=0}^{\lfloor j/3 \rfloor} |B^{1,g}_{n,j}| \right\},$$

then from (21), we obtain

$$\max_{i,j} |u^{k_0}_{i,j}| \leq F_1 G_1 \max_{m,n} |u^{k_0-1}_{m,n}|.$$

Case $k_0 = 2$: Now, after applying two time convolution, we obtain

$$u^{k_0-1}_{m,n} = (u^{k_0-2;0} \star fg)_{m,n} = ((u^{k_0-1;0} \star fg) \star fg)_{i,j}.$$

This implies

$$u^{k_0-1}_{i,j} = \sum_{m=0}^{\lfloor i/3 \rfloor} \sum_{n=0}^{\lfloor j/3 \rfloor} (u^{k_0-1;0} \star fg)_{i,j} f_{i-3m} g_{j-3n}.$$

This leads to

$$u^{k_0-1}_{i,j} = \sum_{m=0}^{\lfloor i/3 \rfloor} \sum_{n=0}^{\lfloor j/3 \rfloor} \left(\sum_{p=0}^{\lfloor m/3 \rfloor} \sum_{s=0}^{\lfloor n/3 \rfloor} u^{k_0-2}_{p,s} f_{m-3p} g_{n-3s} \right) f_{i-3m} g_{j-3n}.$$

This again implies that

$$u^{k_0-1}_{i,j} = \sum_{m=0}^{\lfloor i/3^2 \rfloor} \sum_{n=0}^{\lfloor j/3^2 \rfloor} u^{k_0-2}_{m,n} \sum_{r=3m}^{\lfloor i/3 \rfloor} f_{r-3m} f_{i-3r} \sum_{q=3n}^{\lfloor j/3 \rfloor} g_{q-3n} g_{j-3q}.$$

Further implies

$$u^{k_0-1}_{i,j} = \sum_{m=0}^{\lfloor i/3^2 \rfloor} \sum_{n=0}^{\lfloor j/3^2 \rfloor} u^{k_0-2}_{m,n} \sum_{r=3m}^{\lfloor i/3 \rfloor} B^{1,f}_{m,r} B^{1,f}_{r,i} \sum_{q=3n}^{\lfloor j/3 \rfloor} B^{1,g}_{n,q} B^{1,g}_{q,j}.$$

Furthermore

$$u_{i,j}^{k_0} = \sum_{m=0}^{\lfloor i/3^2 \rfloor} \sum_{n=0}^{\lfloor j/3^2 \rfloor} u_{m,n}^{k_0-2} B_{m,i}^{2,f} B_{n,j}^{2,g}.$$

Now

$$\begin{aligned} \max_{i,j} |u_{i,j}^{k_0}| &= \max_{i,j} \left| \sum_{m=0}^{\lfloor i/3^2 \rfloor} \sum_{n=0}^{\lfloor j/3^2 \rfloor} u_{m,n}^{k_0-2} B_{m,i}^{2,f} B_{n,j}^{2,g} \right|, \\ &\le \max_{i,j} \sum_{m=0}^{\lfloor i/3^2 \rfloor} \sum_{n=0}^{\lfloor j/3^2 \rfloor} \left| B_{m,i}^{2,f} \right| \left| B_{n,j}^{2,g} \right| \max_{m,n} \left| u_{m,n}^{k_0-2} \right|. \end{aligned} \tag{22}$$

Consider

$$F_2 = \max_i \left\{ \sum_{m=0}^{\lfloor i/3^2 \rfloor} \left| B_{m,i}^{2,f} \right| \right\}$$

and

$$G_2 = \max_j \left\{ \sum_{n=0}^{\lfloor j/3^2 \rfloor} \left| B_{n,j}^{2,g} \right| \right\},$$

then we obtain from (22)

$$\max_{i,j} |u_{i,j}^{k_0}| \le F_2 G_2 \max_{m,n} |u_{m,n}^{k_0-2}|.$$

By the same strategy, we get the following reformulations for k_0-th convolution

$$u_{i,j}^{k_0} = (u^{k_0-k_0;0} \star fg)_{m,n} = (\ldots(((u^{k_0-1;0} \star fg) \star fg) \star \ldots \star fg) \star fg)_{i,j}.$$

This implies

$$u_{i,j}^{k_0} = \sum_{m=0}^{\lfloor i/3^{k_0} \rfloor} \sum_{n=0}^{\lfloor j/3^{k_0} \rfloor} u_{m,n}^{0;0} B_{m,i}^{k_0,f} B_{n,j}^{k_0,g} = \sum_{m=0}^{\lfloor i/3^{k_0} \rfloor} \sum_{n=0}^{\lfloor j/3^{k_0} \rfloor} u_{m,n}^{0} B_{m,i}^{k_0,f} B_{n,j}^{k_0,g},$$

where

$$B_{m,i}^{k_0,f} = \sum_{p=3m}^{\lfloor i/3^{k_0-1} \rfloor} B_{m,p}^{k_0-1,f} B_{p,i}^{k_0-1,f}$$

and

$$B_{n,j}^{k_0,g} = \sum_{r=3n}^{\lfloor j/3^{k_0-1} \rfloor} B_{n,s}^{k_0-1,g} B_{n,j}^{k_0-1,g}.$$

Thus,

$$\begin{aligned} \max_{i,j} |u_{i,j}^{k_0}| &= \max_{i,j} \left| \sum_{m=0}^{\lfloor i/3^{k_0} \rfloor} \sum_{n=0}^{\lfloor j/3^{k_0} \rfloor} u_{m,n}^{0} B_{m,i}^{k_0,f} B_{n,j}^{k_0,g} \right|, \\ &\le \max_{i,j} \sum_{m=0}^{\lfloor i/3^{k_0} \rfloor} \sum_{n=0}^{\lfloor j/3^{k_0} \rfloor} \left| B_{m,i}^{k_0,f} \right| \left| B_{n,j}^{k_0,g} \right| \max_{m,n} \left| u_{m,n}^{0} \right|. \end{aligned} \tag{23}$$

Now consider

$$F_{k_0} = \max_i \left\{ \sum_{m=0}^{\lfloor i/3^{k_0} \rfloor} |B_{m,i}^{k_0,f}| \right\} = \max_{i \in \Sigma(k_0,N)} \left\{ \sum_{m=0}^{\lfloor i/3^{k_0} \rfloor} |B_{m,i}^{k_0,f}| \right\} \tag{24}$$

and

$$G_{k_0} = \max_j \left\{ \sum_{n=0}^{\lfloor j/3^{k_0} \rfloor} |B_{n,j}^{k_0,g}| \right\} = \max_{j \in \Sigma(k_0,N)} \left\{ \sum_{n=0}^{\lfloor j/3^{k_0} \rfloor} |B_{n,j}^{k_0,g}| \right\}, \tag{25}$$

then, from (23), we obtain

$$\max_{i,j} |u_{i,j}^{k_0}| \leq F_{k_0} G_{k_0} \max_{m,n} |u_{m,n}^0|,$$

where

$$\max_{i,j} \left\{ \sum_{m=0}^{\lfloor i/3^{k_0} \rfloor} \sum_{n=0}^{\lfloor j/3^{k_0} \rfloor} |B_{m,i}^{k_0,f}||B_{n,j}^{k_0,g}| \right\} = \max_{i,j \in \Sigma(k_0,N)} \left\{ \sum_{m=0}^{\lfloor i/3^{k_0} \rfloor} \sum_{n=0}^{\lfloor j/3^{k_0} \rfloor} |B_{m,i}^{k_0,f}||B_{n,j}^{k_0,g}| \right\}.$$

□

3.1. Subdivision Depth for Ternary Subdivision Surfaces

In this section, we first compute error bounds for subdivision surfaces. Secondly, we use these error bounds to compute subdivision depths by using the methodology given in [21].

Theorem 4. *Consider the initial control polygon $p_{i,j}^0, i, j \in \mathbb{Z}$ and the values $p_{i,j}^k, k \geq 0$, recursively defined by (20) together with (2). Also P^k be the representation of polygon at the points $\{p_{i,j}^k\}$. Then after two consecutive iterations k and $k+1$ the error bounds is given as follows*

$$\|P^{k+1} - P^k\|_\infty \leq (\lambda\beta_1 + \tau\beta_2 + \mu\beta_3)\,(F_{k_0}G_{k_0})^k,$$

where $F_{k_0}, G_{k_0}, k_0 \geq 1$ defined in (24) and (25), $\beta_t = \max_{i,j} \|\triangle_{i,j,t}^0\|, \quad t = 1,2,3,$

$$\triangle_{i,j,1}^k = p_{i+1,j}^k - p_{i,j}^k, \quad \triangle_{i,j,2}^k = p_{i,j+1}^k - p_{i,j}^k, \quad \triangle_{i,j,3}^k = p_{i+1,j+1}^k - p_{i,j+1}^k,$$

where λ, τ and μ are defined in [21].

Proof. See in [21]. □

Theorem 5. *Let a limit surface P^∞ be linked with the subdivision iterative process, then under the same conditions used in Theorem 4 the following inequality hold*

$$\|P^\infty - P^k\|_\infty \leq (\lambda\beta_1 + \tau\beta_2 + \mu\beta_3)\left(\frac{(F_{k_0}G_{k_0})^k}{1 - F_{k_0}G_{k_0}}\right),$$

where $k_0 \geq 1$ is a natural number, such that $F_{k_0}G_{k_0} < 1$.

Proof. See in [21]. □

Remark 2. *Here F_1G_1 is also equal to δ which is defined in [21].*

Theorem 6. *Let k be subdivision depth and let ∇^k be the error bound between ternary subdivision surface P^∞ and its k-level control polygon P^k. For arbitrary $\epsilon > 0$, if*

$$k \geq log_{(F_{k_0}G_{k_0})}\left(\frac{\epsilon(1-F_{k_0}G_{k_0})}{\lambda\beta_1+\tau\beta_2+\mu\beta_3}\right), \tag{26}$$

then $\nabla^k \leq \epsilon$.

Proof. Let ∇^k be the distance between limit surface P^∞ and control polygon P^k at k-th level defined in Theorem 5, such that

$$\nabla^k = \|P^\infty - P^k\|_\infty \leq (\lambda\beta_1+\tau\beta_2+\mu\beta_3)\left(\frac{(F_{k_0}G_{k_0})^k}{1-F_{k_0}G_{k_0}}\right).$$

To obtain given tolerance $\epsilon > 0$, consider

$$(\lambda\beta_1+\tau\beta_2+\mu\beta_3)\left(\frac{(F_{k_0}G_{k_0})^k}{1-F_{k_0}G_{k_0}}\right) \leq \epsilon,$$

which implies

$$\left(\frac{\lambda\beta_1+\tau\beta_2+\mu\beta_3}{\epsilon(1-F_{k_0}G_{k_0})}\right) \leq ((F_{k_0}G_{k_0})^{-1})^k.$$

Now taking logarithm , we have

$$k \geq \frac{\log\left(\frac{\lambda\beta_1+\tau\beta_2+\mu\beta_3}{\epsilon(1-F_{k_0}G_{k_0})}\right)}{\log(F_{k_0}G_{k_0})^{-1}} = \frac{\log\left(\frac{\lambda\beta_1+\tau\beta_2+\mu\beta_3}{\epsilon(1-F_{k_0}G_{k_0})}\right)}{-\log(F_{k_0}G_{k_0})} = -\log_{(F_{k_0}G_{k_0})}\left(\frac{\lambda\beta_1+\tau\beta_2+\mu\beta_3}{\epsilon(1-F_{k_0}G_{k_0})}\right),$$

which implies

$$k \geq \log_{(F_{k_0}G_{k_0})}\left(\frac{\lambda\beta_1+\tau\beta_2+\mu\beta_3}{\epsilon(1-F_{k_0}G_{k_0})}\right)^{-1},$$

which further implies

$$k \geq \log_{(F_{k_0}G_{k_0})}\left(\frac{\epsilon(1-F_{k_0}G_{k_0})}{\lambda\beta_1+\tau\beta_2+\mu\beta_3}\right),$$

then $\nabla^k \leq \epsilon$. This completes the proof. □

3.2. Application for Bivariate Case

Here, we present some numerical examples to compute subdivision depth for subdivision surfaces. The associated constants $F_{k_0}G_{k_0}, k_0 \geq 1$ for some ternary subdivision surfaces by using (24) and (25) are shown in Table 6. We see that the values of $F_{k_0}G_{k_0}$ decrease with the increase of k_0. This is the main advantage of our proposed approach.

Table 6. Associated constants of ternary subdivision surfaces.

Scheme/$F_{k_0}G_{k_0}$	F_1G_1	F_2G_2	F_3G_3	F_4G_4	F_5G_5
2-point scheme [12]	0.333333	0.111111	0.037037	0.012346	0.004115
3-point scheme [30]	0.61716	0.304535	0.145678	0.068303	0.031622
4-point scheme [8]	0.505401	0.233098	0.096089	0.040774	0.016958
4-point scheme [6]	0.54321	0.228729	0.10226	0.047435	0.021475

Example 5. *Given the initial polygon $p^0_{i,j} = p_{i,j}, i, j \in \mathbb{Z}$ with values $p^k_{i,j}, k \geq 1$ be frequent explanation in [12], then the subdivision depths for $F_{k_0}G_{k_0}, k_0 \geq 1$ by using Theorem 6 are shown in Table 7. The first and fifth convolution comparison results are shown in Figure 2a.*

Table 7. Subdivision depth of 2-point ternary subdivision surfaces.

$F_{k_0}G_{k_0}/\epsilon$	2.06×10^{-3}	8.5×10^{-7}	3.49×10^{-9}	1.43×10^{-11}	5.92×10^{-14}	2.43×10^{-16}	1.003×10^{-18}
F_1G_1	5	10	15	20	25	30	35
F_2G_2	3	5	8	10	13	15	18
F_3G_3	1	3	5	6	8	10	11
F_4G_4	1	2	4	5	6	7	9
F_5G_5	1	2	3	4	5	6	7

Example 6. *Given control polygon $p^0_{i,j} = p_{i,j}, i, j \in \mathbb{Z}$ with the values $p^k_{i,j}$ for all positive integers be illustrated by the tensor product of the scheme demonstrated in [30], then the subdivision depths for $F_{k_0}G_{k_0}, k_0 \geq 1$ by using Theorem 6 are shown in Table 8 and in the sense of graphical structure these results are shown in Figure 2b.*

Table 8. Subdivision depth of 3-point ternary subdivision surfaces.

$F_{k_0}G_{k_0}/\epsilon$	7.25×10^{-3}	2.29×10^{-4}	7.25×10^{-6}	2.29×10^{-7}	7.25×10^{-9}	2.29×10^{-10}	7.25×10^{-12}
F_1G_1	9	16	23	31	38	45	52
F_2G_2	3	6	9	12	15	18	21
F_3G_3	2	4	5	7	9	11	13
F_4G_4	1	3	4	5	6	8	9
F_5G_5	1	2	3	4	5	6	7

Example 7. *Given an initial control polygon $p^0_{i,j} = p_{i,j}, i, j \in \mathbb{Z}$ with the values $p^k_{i,j}, k \geq 1$ be frequent explanation by the tensor product of the scheme presented in [8], then the subdivision depths for $F_{k_0}G_{k_0}, k_0 \geq 1$ by using Theorem 6 are shown in Table 9 and graphical results are presented in Figure 2c.*

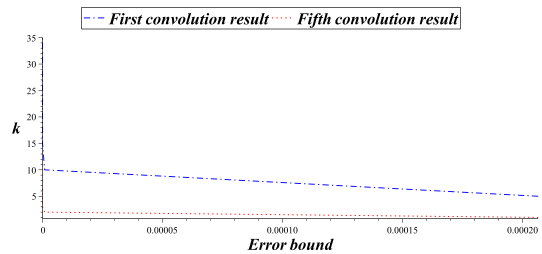
Table 9. Subdivision depth of 4-point ternary subdivision surfaces.

$F_{k_0}G_{k_0}/\epsilon$	6.59×10^{-3}	1.11×10^{-4}	1.18×10^{-6}	3.21×10^{-8}	5.45×10^{-10}	9.24×10^{-12}	1.56×10^{-13}
F_1G_1	7	13	19	25	31	37	43
F_2G_2	3	6	9	11	14	17	20
F_3G_3	2	4	5	7	9	10	12
F_4G_4	1	3	4	5	6	8	9
F_5G_5	1	2	3	4	5	6	7

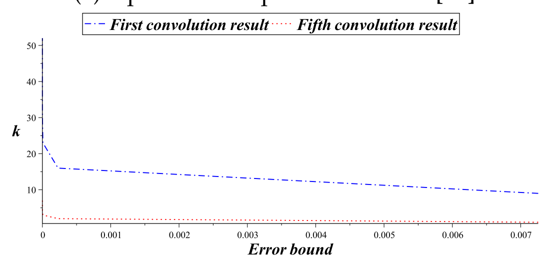
Example 8. *Given $p^0_{i,j} = p_{i,j}, i, j \in \mathbb{Z}$ be the initial polygon with values $p^k_{i,j}, k \geq 1$ be illustrated by the tensor product of the scheme presented in [6] with $w = 1/12$, then the subdivision depths for $F_{k_0}G_{k_0}, k_0 \geq 1$ are shown in Table 10. Also demonstration of graphical view are given in Figure 2d.*

Table 10. Subdivision depth of 4-point ternary subdivision surfaces.

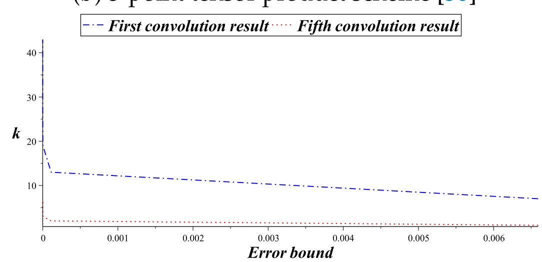
$F_{k_0}G_{k_0}/\epsilon$	$\mathbf{7.31 \times 10^{-3}}$	$\mathbf{1.57 \times 10^{-4}}$	$\mathbf{3.37 \times 10^{-6}}$	$\mathbf{7.24 \times 10^{-8}}$	$\mathbf{1.55 \times 10^{-9}}$	$\mathbf{3.34 \times 10^{-11}}$	$\mathbf{7.17 \times 10^{-13}}$
F_1G_1	8	14	20	26	33	39	45
F_2G_2	3	5	8	11	13	16	18
F_3G_3	2	3	5	7	8	10	12
F_4G_4	1	3	4	5	6	8	9
F_5G_5	1	2	3	4	5	6	7



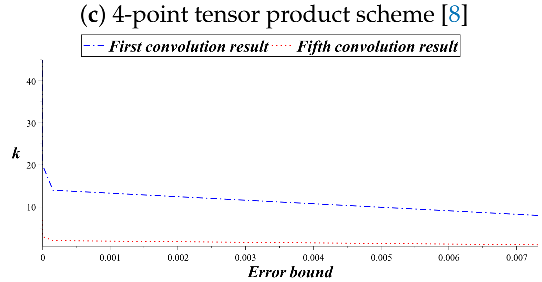
(**a**) 2-point tensor product scheme [12]



(**b**) 3-point tensor product scheme [30]



(**c**) 4-point tensor product scheme [8]



(**d**) 4-point tensor product scheme [6]

Figure 2. Comparison between first and fifth convolutions. This shows error decreases with the increase of convolution. Of course, it decreases with the increase of subdivision depth.

4. Conclusions

We described a formula to find the sharp error bounds between the polygon at any stage and the limiting polygon of the subdivision scheme. In addition, we have achieved a computational formula of subdivision depth for ternary subdivision schemes by using the error bounds. Existing methods only work under the strong condition given in ([21], Equation (2.3)). In this paper, we relaxed the strict condition by convolving the mask of the schemes. Using our framework, we can get sharp bounds and subdivision depths by increasing the convolution steps. Ultimately, the suggested numerical method work when the other methods fail. In addition, extensive numerical experiments predict that if we have a prescribed error tolerance then a finer shape can be obtained by using fewer subdivision steps (i.e., depths). In the future, we will generalize our framework for higher arity (i.e., quaternary, quinary, and so on) schemes.

Author Contributions: Conceptualization, F.K. and G.M.; formal analysis, A.S. and D.B.; methodology, D.B. and M.M.A.-Q.; supervision, D.B.; writing–original draft, F.K., G.M. and A.S.; writing–review and editing, D.B. and M.M.A.-Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: We would like to express our sincere thanks to anonymous reviewers for the careful reading and helpful remarks on our paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ju, C.; Solomonik, E. Derivation and analysis of fast bilinear algorithms for convolution. *arXiv* **2019**, arXiv:1910.13367v1.
2. Aslam, M.; Mustafa, G.; Ghaffar, A. (2n-1)-point ternary approximating and interpolating subdivision schemes. *J. Appl. Math.* **2011**, *11*, 1–12.
3. Faheem, K.; Mustafa, G. Ternary six-point interpolating subdivision scheme. *Lobachevskii J. Math.* **2008**, *29*, 153–163.
4. Ghaffar, A.; Mustafa, G. The family of even-point ternary approximating schemes. *ISRN Appl. Math.* **2012**, *12*, 1–14.
5. Mustafa, G.; Ghaffar, A.; Khan, F. The odd-point ternary approximating schemes. *Am. J. Comput. Math.* **2011**, *11*, 111–118.
6. Hassan, M.F.; Ivrissimitzis, I.P.; Dodgson, N.A.; Sabin, M.A. An interpolating 4-point C^2 ternary stationary subdivision scheme. *Comput. Aided Geom. Des.* **2002**, *19*, 1–18.
7. Siddiqi, S.S.; Rehan, K. Symmetric ternary interpolating C^1 subdivision scheme. *Int. Math. Forum* **2012**, *7*, 2269–2277.
8. Kwan, P.K.; Byung, G.L.; Gang, J.Y. A ternary 4-point approximating subdivision scheme. *Appl. Math. Comput.* **2007**, *190*, 1563–1573.
9. Mustafa, G.; Ashraf, P. A new 6-point ternary interpolating subdivision scheme and its differentiability. *J. Inf. Comput. Sci.* **2010**, *5*, 199–210.
10. Mustafa, G.; Irum, J.; Bari, M. A new 5-point ternary interpolating subdivision scheme and its differentiability. *ISRN Comput. Math.* **2012**, *12*, 1–10.
11. Siddiqi, S.S.; Ahmed, N.; Rehan, K. Ternary 4-point interpolating scheme for curve sketching. *Res. J. Appl. Sci. Eng. Technol.* **2013**, *6*, 1556–1561.
12. Siddiqi, S.S.; Younis, M. Construction of ternary approximating subdivision schemes. *UPB Sci. Bull. Ser. A Appl. Math. Phys.* **2014**, *76*, 71–78.
13. Peng, K.; Tan, J.; Li, Z.; Zhang, L. Fractal behavior of a ternary 4-point rational interpolation subdivision scheme. *Math. Comput. Appl.* **2018**, *23*, 1–10.

14. Song, Q.; Zheng, H.C.; Peng, G.H. A nonlinear ternary circle-preserving interpolatory subdivision scheme. *Appl. Mech. Mater.* **2013**, *427*, 2170–2173.
15. Aslam, M. A family of 5-point nonlinear ternary interpolating subdivision schemes with C^2 smoothness. *Math. Comput. Appl.* **2018**, *23*, 1–8.
16. Beccari, C.; Casciola, G.; Romani, L. An interpolating 4-point C^2 ternary non-stationary subdivision scheme with tension control. *Comput. Aided Geom. Des.* **2007**, *24*, 210–219.
17. Cai, Z. Convergence, error estimation and some properties of four-point interpolation subdivision scheme. *Comput. Aided Geom. Des.* **1995**, *12*, 459–468.
18. Conti, C.; Jetter, K. Concerning order of convergence for subdivision. *Numer. Algorithms* **2004**, *36*, 345–363.
19. Huning, S.; Wallner, J. Convergence of subdivision schemes on Riemannian manifolds with nonpositive sectional curvature. *Adv. Comput. Math.* **2019**, *45*, 1689–1709.
20. Mustafa, G.; Chen, F.; Deng, J. Estimating error bounds for binary subdivision curves/surfaces. *J. Comput. Appl. Math.* **2006**, *193*, 596–613.
21. Mustafa, G.; Deng, J. Estimating error bounds for ternary subdivision curve/surfaces. *J. Comput. Math.* **2007**, *25*, 473–484.
22. Mustafa, G.; Hashmi, S.; Noshi, N.A. Estimating error bounds for tensor product binary subdivision volumetric model. *Int. J. Comput. Math.* **2006**, *83*, 879–903.
23. Mustafa, G.; Hashmi, S.; Faheem, K. Estimating error bounds for non-stationary binary subdivision curves/surfaces. *J. Inf. Comput. Sci.* **2007**, *2*, 179–190.
24. Moncayo, M.; Amat, S. Error bounds for a class of subdivision schemes based on the two-scale refinement equation. *J. Comput. Appl. Math.* **2011**, *236*, 265–278.
25. Deng, C.; Jin, W.; Li, Y.; Xu, H. A formula for estimating the deviation of a binary interpolatory subdivision curve from its polygon. *Appl. Math. Comput.* **2017**, *304*, 10–19.
26. Mustafa, G.; Hashmi, S. Subdivision depth computation for *n*-ary subdivision curves/surfaces. *Vis. Comput.* **2010**, *26*, 841–851.
27. Mustafa, G. Subdivision depth computation for tensor product ternary volumetric model. In Proceedings of the 2009 International Conference on Scientific Computing, CSC 2009, Las Vegas, NV, USA, 13–16 July 2009; pp. 151–158.
28. Mustafa, G.; Hashmi, S.; Faheem, K. Subdivision depth for triangular surfaces. *Alex. Eng. J.* **2016**, *55*, 1647–1653.
29. Shahzad, A.; Faheem, K.; Ghaffar, A.; Mustafa, G.; Nisar, K.S.; Baleanu, D. A novel numerical algorithm to estimate the subdivision depth of binary subdivision schemes. *Symmetry* **2020**, *12*, 66.
30. Hassan, M.F.; Dodgson, N.A. *Ternary and 3-Point Univariate Subdivision Scheme*; Cohen, A., Marrien, J.L., Schumaker, L.L., Eds.; Curve and Surface fitting: Sant-Malo 2002; Nashboro Press: Brentwood, UK, 2003; pp. 199–208.