



# Article A Novel Numerical Method for Computing Subdivision Depth of Quaternary Schemes

Aamir Shahzad<sup>1</sup>, Faheem Khan<sup>1</sup>, Abdul Ghaffar<sup>2</sup>, Shao-Wen Yao<sup>3,\*</sup>, Mustafa Inc<sup>4,5,6,\*</sup>, and Shafqat Ali<sup>7</sup>

- <sup>1</sup> Department of Mathematics, University of Sargodha, Sargodha 40100, Pakistan; uos.amir@gmail.com (A.S.); fahimscholar@gmail.com (F.K.)
- <sup>2</sup> Department of Mathematics, Ghazi University D G Khan, D G Khan 32200, Pakistan; abdulghaffar.jaffar@gmail.com
- <sup>3</sup> School of Mathematics and Information Science, Henan Polytechnic University, Jiaozuo 454000, China
- <sup>4</sup> Department of Computer Engineering, Biruni University, Istanbul 34096, Turkey
- <sup>5</sup> Department of Mathematics, Science Faculty, Firat University, Elazig 23119, Turkey
- <sup>6</sup> Department of Medical Research, China Medical University Hospital, China Medical University, Taichung 40402, Taiwan
- <sup>7</sup> Department of Mathematics, The Islamia University of Bahawalpur, Bahawalpur 63100, Pakistan; shafqat.ali@iub.edu.pk
- \* Correspondence: yaoshaowen@hpu.edu.cn (S.-W.Y.); minc@firat.edu.tr (M.I.)

**Abstract:** In this paper, an advanced computational technique has been presented to compute the error bounds and subdivision depth of quaternary subdivision schemes. First, the estimation is computed of the error bound between quaternary subdivision limit curves/surfaces and their polygons after *k*th-level subdivision by using  $l_0$  order of convolution. Secondly, by using the error bounds, the subdivision depth of the quaternary schemes has been computed. Moreover, this technique needs fewer iterations (subdivision depth) to get the optimal error bounds of quaternary subdivision schemes as compared to the existing techniques.

**Keywords:** quaternary subdivision scheme; subdivision models; inequalities; convolution; error bound; subdivision depth; curves and surfaces

MSC: 65D17; 65D05; 65U07

#### 1. Introduction

Subdivision schemes are major tools in Geometric Modeling. These tools are mainly used to produce curve and surface models. The schemes are categorized into binary, ternary, quaternary, . . ., n-ary schemes. Presently, thousands of schemes have been introduced in each category. All these schemes help the technologist to produce the refine models to meet the requirements of the investors in the area of engineering. The initial sketch and subdivision rules are the main ingredients of these schemes. The estimation of the error bounds of the limit models from its initial sketch is one of the important tasks. Another task is to find the number of subdivision steps (depth) required to get the user-defined tolerable error. These two tasks are also called the distance/error between the limit model & its *k*th level model and subdivision depth, respectively. In this paper, we address these tasks for quaternary schemes. First, we give an overview of quaternary subdivision schemes (QSS) before addressing these tasks.

In general, QSS has four rules to refine each edge of the initial polygon (sketch). These rules are the affine combination of the points of the polygon, and they produce successively refined sketches. In the limiting case, we get the limiting model. Initially, Mustafa and Faheem [1] introduced 4-point approximating QSS which produces  $C^3$  models. The generalized idea of *m*-point approximating QSS is given by Siddiqi and Younis [2]. They also introduced interpolating QSS in the same year [3]. A 4-point QSS is presented by



Citation: Shahzad, A.; Khan, F.; Ghaffar, A.; Yao, S.-W.; Inc, M.; Ali, A. A Novel Numerical Method for Computing Subdivision Depth of Quaternary Schemes. *Mathematics* 2021, *9*, 809. https://doi.org/ 10.3390/math9080809

Received: 19 February 2021 Accepted: 1 April 2021 Published: 8 April 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). Pervaiz [4] in 2018. Moreover, the QSS also belongs to the classes of the schemes introduced by [5–13] in different years. So, the importance of the QSS cannot be denied. Furthermore, the tasks of finding the error bounds and subdivision depth of models produced by QSS are meaningful.

The first technique was introduced by Mustafa et al. [14] in 2006, then its generalized version for QSS was presented by Mustafa et al. [15] in 2010. The further generalization has also been done for other categories of the schemes [16]. This technique is not suitable to use for some subdivision schemes. We also mention the drawback of this technique in this paper. The second technique is introduced by Deng et al. [17]. It is not mature enough. It only works for binary interpolating schemes. Its generalization to the cases of *n*-ary subdivision schemes needs to be investigated.

The third technique is introduced by Moncayo and Amat [18] and Shahzad et al. [19]. It works for binary class of schemes. Its generalization for the ternary class of schemes was introduced by Faheem et al. [20]. In this work, we are interested in generalizing the technique for QSS.

The remaining part of the work is configured as follows: In Sections 2 and 3, we present general inequalities to compute the error bound and subdivision depth of curve and surface models produced by QSS, respectively. In Sections 4 and 5, we offer the applications of these inequalities for curve and surface models, respectively. The conclusion will be drawn in Section 6.

#### 2. The Error Bounds and Subdivision Depth for Curve Models

If the sequence of points  $\{p_i^k; i \in \mathbb{Z}\}$  show a succession in  $\mathbb{R}^{\aleph}$ , where  $\aleph \ge 2$  and the index  $k \ge 0$  represents the subdivision level (number of iterations) then the configuration of the (k + 1)th level points computed by QSS is shown in Figure 1. A generalized mathematical form of the QSS is presented as the affine combination of the points [15],

$$p_{4i+\alpha}^{k+1} = \sum_{m=0}^{N-1} a_{\alpha,m} p_{i+m}^k, \quad \alpha = 0, 1, 2, 3.$$
(1)

Since the combination is affine it holds

$$\sum_{m=0}^{N-1} a_{\alpha,m} = 1, \quad \alpha = 0, 1, 2, 3.$$
<sup>(2)</sup>

The adjustment of the coefficients for the computation of error bound and subdivision depth is

$$\begin{cases} e_{\beta,m} = \sum_{l=0}^{m} (a_{\beta,l} - a_{\beta+1,l}), & \beta = 0, 1, 2, \\ e_{3,m} = a_{0,m} - \sum_{\beta=0}^{2} e_{\beta,m}, \end{cases}$$
(3)

along with the strict condition (See [15])

$$\sum_{m=0}^{N-1} |e_{\beta,m}| < 1, \quad \beta = 0, 1, 2, 3.$$
(4)



**Figure 1.** The configuration of new points for the interpolating and approximating QSS for curves; the solid lines are the control polygon, and the dotted lines are the refined polygon.

Here, we introduce some new notations, for m = 0, 1, ..., N - 1, as follows

$$b_{4m} = e_{0,m},$$
  

$$b_{4m+1} = e_{1,m},$$
  

$$b_{4m+2} = e_{2,m},$$
  

$$b_{4m+3} = e_{3,m}.$$
(5)

Furthermore, to be more specific, update the track defined in [20] for the computation of error bounds and subdivision depth. Let, at the *k*th level of resolution, the vector  $v_i = p_i^k$  represents the approximation coefficients. Then the approximation coefficients of two consecutive stages *k* and *k* + 1 in the reconstruction process of QSS is defined as

$$p_i^{k+1} = \sum_{n \in \mathbb{N}} b_{i-4n} p_n^k = (p^{k;0} \star b)_i,$$
(6)

where  $v_i^0 = p_i^{k;0}$  shows the  $k^{th}$  resolution level and ' $\star$ ' shows the convolution product. If  $p^{k;0} = \{p^{k;0}; n \ge 0\}$  and  $b = \{b_n; n \ge 0\}$  are finite vectors with length  $\gamma_1$  and  $\gamma_2$  respectively then

$$(p^{k;0} \star b)_j = \sum_{n=\max\{j-(\gamma_2-1),0\}}^{\min\{j,\gamma_1-1\}} p_n^{k;0} b_{j-4n}, j = 0, 1, \dots, \gamma_1 + \gamma_1 - 2.$$
(7)

Now we move forward and present some results of successive convolutions for onedimensional array of vectors based on QSS.

**Lemma 1.** Let  $p = \{p_n\}_{n\geq 0}$  and  $b = \{b_n\}_{n=0}^{4N-1}$  with  $b_n = 0$  for  $n \geq 4N$  be finite onedimensional arrays of vectors. Then for QSS, the one-dimensional  $l_0$  convolution of these vectors satisfies the following inequality

$$\|((\dots(((p^{(0)} \star b)^{(0)}) \star b)^{(0)} \star \dots \star b)^{(0)} \star b)\|_{\infty} \le \|p\|_{\infty} \max_{j} \left\{ \sum_{m=0}^{\lfloor j/4^{l_{0}} \rfloor} |A_{m,j}^{l_{0}}| \right\},$$
(8)

where

$$\begin{cases}
A_{m,j}^{1} = b_{j-4m}, \\
A_{m,j}^{l_{0}} = \sum_{p=4m}^{\lfloor j/4^{l_{0}-1} \rfloor} A_{m,p}^{1} A_{p,j}^{l_{0}-1}, \quad l_{0} \ge 2,
\end{cases}$$
(9)

and

$$j \in \Sigma(l_0, N) = \{ \Omega(l_0, N) - 4^{l_0} + 1, \Omega(l_0, N) - 4^{l_0} + 2, \dots, \Omega(l_0, N) \},$$

$$\Omega(l_0, N) = (4^{l_0} - 3)(4N - 1).$$

$$(10)$$

**Proof.** See Appendix A.1.  $\Box$ 

**Lemma 2.** For QSS, the term  $A_{m,i}^{l_0}$  defined by (9) satisfies the following equality

$$A_{m-1,j-4^{l_0}}^{l_0} = A_{m,j}^{l_0} = A_{m+1,j+4^{l_0}}^{l_0}.$$
 (12)

**Proof.** See Appendix A.2.  $\Box$ 

**Corollary 1.** The term  $\max_{j} \left\{ \sum_{m=0}^{\lfloor j/4^{l_0} \rfloor} |A_{m,j}^{l_0}| \right\}$  involved in the inequality (8) satisfies the following equality

$$B_{l_0} = \max_{j} \left\{ \sum_{m=0}^{\lfloor j/4^{l_0} \rfloor} |A_{m,j}^{l_0}| \right\} = \max_{j \in \Sigma(l_0,N)} \left\{ \sum_{m=0}^{\lfloor j/4^{l_0} \rfloor} |A_{m,j}^{l_0}| \right\}.$$
 (13)

**Proof.** See Appendix A.3.  $\Box$ 

Now, we present the inequalities to compute the error bound and subdivision depth for the curve models produced by QSS.

**Theorem 1.** If  $P^0 = \{p_i^0, i \in \mathbb{Z}\}$  is the initial polygon and  $P^k = \{p_i^k, i \in \mathbb{Z}\}$  is the polygon obtained by QSS at  $k^{th}$  subdivision level. Then the error bound between two successive levels is

$$\|P^{k+1} - P^k\|_{\infty} \le \psi \chi(B_{l_0})^k,\tag{14}$$

where  $B_{l_0}$ ,  $l_0 \ge 1$  defined in (13),

$$\chi = \max_{i} \left\| p_{i+1}^{0} - p_{i}^{0} \right\|, \psi = \max_{\alpha} \left( \left| \sum_{m=0}^{N-2} \tilde{a}_{\alpha,m} \right|, \alpha = 0, 1, 2, 3 \right),$$

and

$$\begin{cases} \tilde{a}_{\alpha,0} = \sum_{l=1}^{N-1} a_{\alpha,l} - \frac{\alpha}{4}, \\ \\ \tilde{a}_{\alpha,m} = \sum_{l=m+1}^{N-1} a_{\alpha,l}, \quad m \ge 1, \quad \alpha = 0, 1, 2, 3. \end{cases}$$

We omit the proof since it is similar to the one given in [15].

**Theorem 2.** If we assume the same conditions as in the Theorem 1 with the limiting curve model  $P^{\infty}$  then the error bounds  $\nabla^k$  between the limiting curve model and its  $k^{th}$  level polygon satisfies the inequality

$$\nabla^{k} = \left\| P^{\infty} - P^{k} \right\|_{\infty} \le \psi \chi \left( \frac{(B_{l_0})^{k}}{1 - B_{l_0}} \right), \tag{15}$$

where  $l_0 \ge 1$ , such that  $B_{l_0} < 1$ .

By looking at the proof of Theorem 2.1 of [15] one may lead to prove of Theorem 2. Given a user tolerable error  $\epsilon > 0$ , the subdivision depth of limiting model  $P^{\infty}$  generated

by QSS concerning for  $\epsilon$  is a positive integer k such that the error bound  $\nabla^k \leq \epsilon$ . In the following theorem, we compute the subdivision depth.

**Theorem 3.** *If we assume the same conditions as in the Theorem 2 with the user tolerable error*  $\epsilon > 0$  *and* 

$$k \ge \log_{B_{l_0}}\left(\frac{\epsilon(1-B_{l_0})}{\psi\chi}\right),\tag{16}$$

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

## 3. The Error Bounds and Subdivision Depth for Surface Models

In this work, we first generalize the results presented in Lemma 1 to Lemma 2 and Corollary 1 for two-dimensional arrays then we generalize the inequalities of Theorems 1–3 to compute the error bound and subdivision depth of the limiting tensor product surface models generated by QSS.

For this, let  $\mathbf{P}^k = \{p_{i,j}^k; i, j \in \mathbb{Z}\}$  be the polygon made by the sequence of points in  $\mathbf{R}^{\aleph}$ , where  $\aleph \ge 2$  and the polygon  $\mathbf{P}^{k+1} = \{p_{i,j}^{k+1}; i, j \in \mathbb{Z}\}$  be obtained by the tensor product of the scheme (1). The graphical representation of the points at *k*th and (k + 1)th levels is shown in Figure 2 whereas the mathematical form of tensor product QSS is described as

$$p_{4i+\alpha,4j+\gamma}^{k+1} = \sum_{r=0}^{N-1} \sum_{s=0}^{N-1} a_{\alpha,r} a_{\gamma,s} p_{i+r,m+s}^k, \quad \alpha, \gamma = 0, 1, 2, 3,$$
(17)

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



**Figure 2.** The configuration of new points for the interpolating and approximating QSS for the surface. The solid lines show one face of coarse polygon while the dotted lines show the faces of the refined polygon.

Here we introduce new notations  $c = \{c_n\}_{n \in \mathbb{N}}$ ,  $d = \{d_n\}_{n \in \mathbb{N}}$  for r, s = 0, 1, ..., N - 1, such that

$$c_{4r} = a_{0,N-r-1},$$

$$c_{4r+1} = a_{1,N-r-1},$$

$$c_{4r+2} = a_{2,N-r-1},$$

$$c_{4r+3} = a_{3,N-r-1},$$

$$r = 0, ..., N - 1.$$

$$d_{4s} = b_{0,N-s-1},$$

$$d_{4s+1} = b_{1,N-s-1},$$

$$d_{4s+2} = b_{2,N-s-1},$$

$$d_{4s+3} = b_{3,N-s-1},$$

$$s = 0, ..., N - 1.$$
(18)

Since the extension of some of the results from one dimension array of vectors to twodimensional array is straight forward (See [15]), therefore we skip the trivial explanations and directly come to the following result.

**Lemma 3.** Let  $p_{i,j}$  be a finite two-dimensional array of vectors and  $d = \{d_n\}_{n=0}^{4N-1}, c = \{c_n\}_{n=0}^{4N-1}$  with  $d_n = c_n = 0$  for  $n \ge 4N$  are one-dimensional arrays of vectors. Then, for QSS, the two-dimensional  $l_0$  convolution satisfies the following inequality

$$\max_{i,j} |p_{i,j}^{l_0}| \le C_{l_0} D_{l_0} \max_{m,n} |p_{m,n}^0|.$$
(19)

Here

$$C_{l_0} = \max_{i} \left\{ \sum_{m=0}^{\lfloor i/4^{l_0} \rfloor} |A_{m,i}^{l_0,c}| \right\}$$
(20)

and

$$D_{l_0} = \max_{j} \left\{ \sum_{n=0}^{\lfloor j/4^{l_0} \rfloor} |A_{n,j}^{l_0,d}| \right\}.$$
 (21)

where

$$\begin{cases} A_{m,i}^{1,c} = c_{i-4m}, \\ A_{m,i}^{l_0,c} = \sum_{p=4m}^{\lfloor i/4^{l_0-1} \rfloor} A_{m,p}^{1,c} A_{p,i}^{l_0-1,c}, \\ A_{n,j}^{1,d} = d_{j-4n}, \\ A_{n,j}^{l_0,d} = \sum_{q=4n}^{\lfloor j/4^{l_0-1} \rfloor} A_{n,q}^{1,d} A_{q,j}^{l_0-1,d}, \qquad l_0 \ge 2, \end{cases}$$

**Proof.** See Appendix A.4.  $\Box$ 

Now, we present the inequalities to compute the error bound and subdivision depth for the surface models produced by QSS. By using a similar approach of [15], one can easily prove the Theorems 4 and 5.

**Theorem 4.** If  $\mathbf{P}^0 = \{p_{i,j'}^0, i \in \mathbb{Z}\}$  is the initial polygon and  $\mathbf{P}^k = \{p_{i,j'}^k, i \in \mathbb{Z}\}$  is the polygon obtained by (17). Then the error bound between two successive levels is

$$\|\boldsymbol{P}^{k+1} - \boldsymbol{P}^k\|_{\infty} \le \nu (C_{l_0} D_{l_0})^k.$$
(22)

Here  $C_{l_0}$ ,  $D_{l_0}$  for  $l_0 \ge 1$  is defined in (20) and (21) and  $\nu = \max_{\alpha,\beta} \left\{ \sum_{t=1}^{3} (\chi_t)(\psi_{\alpha,\beta}^t), \alpha, \beta = 0, 1, 2, 3 \right\}$ , where  $\chi_t$  and  $\psi_{\alpha,\beta}^t$  for  $\alpha, \beta = 0, 1, 2, 3$  are defined in [15].

**Theorem 5.** If we assume the same conditions as in the Theorem 4 with the limiting surface model  $P^{\infty}$  then the error bound  $\nabla^k$  between the limiting surface model and its kth level polygon is defined by the inequality

$$\nabla^{k} = \| \mathbf{P}^{\infty} - \mathbf{P}^{k} \|_{\infty} \le \nu \left( \frac{(C_{l_{0}} D_{l_{0}})^{k}}{1 - C_{l_{0}} D_{l_{0}}} \right),$$
(23)

where  $l_0 \geq 1$ , such that  $C_{l_0}D_{l_0} < 1$ .

In the following theorem, we present the subdivision depth for the surface model.

**Theorem 6.** *If we assume the same conditions as in the Theorem 5 with the user tolerable error*  $\epsilon > 0$  *and if* 

$$k \ge \log_{(C_{l_0}D_{l_0})}\left(\frac{\epsilon(1-C_{l_0}D_{l_0})}{\nu}\right),$$
 (24)

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

## 4. Numerical Applications for Curve Models

In this section, we demonstrate the performance of our inequalities to compute error bound and subdivision depth of the curve models. First, we compute  $B_{l_0}$  defined in (13) at different values of  $l_0$ .

**Example 1.** If the curve model is produced by a 3-point approximating QSS [2] with coefficients  $a_{0,0} = \frac{49}{128}, a_{0,1} = \frac{39}{64}, a_{0,2} = \frac{1}{128}, a_{1,0} = \frac{25}{128}, a_{1,1} = \frac{47}{64}, a_{1,2} = \frac{9}{128}, a_{2,0} = \frac{9}{128}, a_{2,1} = \frac{47}{64}, a_{2,2} = \frac{25}{128}, a_{3,0} = \frac{1}{128}, a_{3,1} = \frac{39}{64}, a_{3,2} = \frac{49}{128}$ . Then for N = 3, we have

$$B_{l_0} = \max_{j \in \Sigma(l_0,3)} \bigg\{ \sum_{m=0}^{\lfloor j/4^{l_0} \rfloor} |A_{m,j}^{l_0}| \bigg\}.$$

For  $l_0 = 1$ , we get

$$B_{1} = \max_{j \in \Sigma(1,3)} \left\{ \sum_{m=0}^{\lfloor j/4 \rfloor} |A_{m,j}^{1}| \right\} = \max_{j \in \{8,9,10,11\}} \left\{ \sum_{m=0}^{\lfloor j/4 \rfloor} |b_{j-4m}| \right\}.$$

Using (5) and Lemma 1, we have  $b = \{b_n\}_{n=0}^{11}$  with  $b_n = 0$  for  $n \ge 12$ . Hence

$$\{b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{11}\} = \left\{\frac{24}{128}, \frac{16}{128}, \frac{8}{128}, \frac{1}{128}, \frac{8}{128}, \frac{16}{128}, \frac{24}{128}, \frac{30}{128}, 0, 0, 0, \frac{1}{128}\right\}.$$
 (25)

Now consider

$$B_{1} = \max\left\{\sum_{m=0}^{\lfloor 8/4 \rfloor} |b_{8-4m}|, \sum_{m=0}^{\lfloor 9/4 \rfloor} |b_{9-4m}|, \sum_{m=0}^{\lfloor 10/4 \rfloor} |b_{10-4m}|, \sum_{m=0}^{\lfloor 11/4 \rfloor} |b_{11-4m}|\right\}$$

This implies

$$B_{1} = \max\left\{ |b_{8}| + |b_{4}| + |b_{0}|, |b_{9}| + |b_{5}| + |b_{1}|, |b_{10}| + |b_{6}| + |b_{2}|, |b_{11}| + |b_{7}| + |b_{3}| \right\}$$
  
$$= \max\left\{ 0 + \left| \frac{8}{128} \right| + \left| \frac{24}{128} \right|, 0 + \left| \frac{16}{128} \right| + \left| \frac{16}{128} \right|, 0 + \left| \frac{24}{128} \right| + \left| \frac{8}{128} \right|, \left| \frac{1}{128} \right| + \left| \frac{30}{128} \right| + \left| \frac{1}{128} \right| \right\}$$
  
$$= \frac{1}{4}.$$

For  $l_0 = 2$ , we get

$$B_{2} = \max_{j \in \Sigma(2,3)} \left\{ \sum_{m=0}^{\lfloor j/4^{2} \rfloor} |A_{m,j}^{2}| \right\} = \max_{j \in \{128,129,\dots,143\}} \left\{ \sum_{m=0}^{\lfloor j/16 \rfloor} |A_{m,j}^{2}| \right\}$$
$$= \max_{j \in \{128,129,\dots,143\}} \left\{ \sum_{m=0}^{\lfloor j/16 \rfloor} \left| \sum_{n=4m}^{\lfloor j/4 \rfloor} A_{m,n}^{1} A_{n,j}^{1} \right| \right\}.$$

This implies

$$B_{2} = \max\left\{ \sum_{m=0}^{\lfloor 128/16 \rfloor} \left| \sum_{n=4m}^{\lfloor 128/4 \rfloor} A_{m,n}^{1} A_{n,128}^{1} \right|, \sum_{m=0}^{\lfloor 129/16 \rfloor} \left| \sum_{n=4m}^{\lfloor 129/4 \rfloor} A_{m,n}^{1} A_{n,129}^{1} \right|, \sum_{n=4m}^{\lfloor 130/16 \rfloor} A_{m,n}^{1} A_{n,129}^{1} \right|, \sum_{m=0}^{\lfloor 130/16 \rfloor} A_{m,n}^{1} A_{n,130}^{1} \right|, \sum_{m=0}^{\lfloor 131/16 \rfloor} A_{m,n}^{1} A_{n,131}^{1} \right|, \sum_{m=0}^{\lfloor 132/16 \rfloor} \left| \sum_{n=4m}^{\lfloor 132/4 \rfloor} A_{m,n}^{1} A_{n,132}^{1} \right|, \sum_{m=0}^{\lfloor 133/16 \rfloor} \left| \sum_{n=4m}^{\lfloor 133/4 \rfloor} A_{m,n}^{1} A_{n,131}^{1} \right|, \sum_{m=0}^{\lfloor 135/16 \rfloor} \left| \sum_{n=4m}^{\lfloor 135/4 \rfloor} A_{m,n}^{1} A_{n,132}^{1} \right|, \sum_{m=0}^{\lfloor 133/16 \rfloor} \left| \sum_{n=4m}^{\lfloor 133/4 \rfloor} A_{m,n}^{1} A_{n,133}^{1} \right|, \sum_{m=0}^{\lfloor 135/4 \rfloor} A_{m,n}^{1} A_{n,132}^{1} \right|, \sum_{m=0}^{\lfloor 135/4 \rfloor} A_{m,n}^{1} A_{n,132}^{1} \right|, \sum_{m=0}^{\lfloor 135/4 \rfloor} A_{m,n}^{1} A_{n,135}^{1} \right|, \sum_{m=0}^{\lfloor 135/4 \rfloor} A_{m,n}^{1} A_{n,136}^{1} \right|, \sum_{m=0}^{\lfloor 135/4 \rfloor} A_{m,n}^{1} A_{n,137}^{1} \right|, \sum_{m=0}^{\lfloor 138/4 \rfloor} \left| \sum_{n=4m}^{\lfloor 138/4 \rfloor} A_{m,n}^{1} A_{n,138}^{1} \right|, \sum_{m=0}^{\lfloor 139/16 \rfloor} \left| \sum_{n=4m}^{\lfloor 139/4 \rfloor} A_{m,n}^{1} A_{n,139}^{1} \right|, \sum_{m=0}^{\lfloor 138/4 \rfloor} A_{m,n}^{1} A_{n,137}^{1} \right|, \sum_{m=0}^{\lfloor 141/16 \rfloor} \left| \sum_{n=4m}^{\lfloor 138/4 \rfloor} A_{m,n}^{1} A_{n,136}^{1} \right|, \sum_{m=0}^{\lfloor 141/4 \rfloor} A_{m,n}^{1} A_{n,141}^{1} \right|, \sum_{m=0}^{\lfloor 142/4 \rfloor} A_{m,n}^{1} A_{n,142}^{1} \right|, \sum_{m=0}^{\lfloor 143/16 \rfloor} \left| \sum_{n=4m}^{\lfloor 143/16 \rfloor} A_{m,n}^{1} A_{n,141}^{1} \right|, \sum_{m=0}^{\lfloor 143/16 \rfloor} \left| \sum_{n=4m}^{\lfloor 143/4 \rfloor} A_{m,n}^{1} A_{n,142}^{1} \right|, \sum_{m=0}^{\lfloor 143/16 \rfloor} \left| \sum_{n=4m}^{\lfloor 143/4 \rfloor} A_{m,n}^{1} A_{n,143}^{1} \right|, \sum_{m=0}^{\lfloor 143/16 \rfloor} \left| \sum_{n=4m}^{\lfloor 143/4 \rfloor} A_{m,n}^{1} A_{n,143}^{1} \right|, \sum_{m=0}^{\lfloor 143/16 \rfloor} \left| \sum_{n=4m}^{\lfloor 143/4 \rfloor} A_{m,n}^{1} A_{n,143}^{1} \right|, \sum_{m=0}^{\lfloor 143/4 \rfloor} A_{m,n}^{1} A_{n,143}^{1} \right|, \sum_{m=0}^{\lfloor 143/16 \rfloor} \left| \sum_{n=4m}^{\lfloor 143/4 \rfloor} A_{m,n}^{1} A_{n,143}^{1} \right|, \sum_{m=0}^{\lfloor 143/4 \rfloor} A_{m,n}^{1} A_{m,143}^{1} A_{m,n}^{1} A_{m,143}^{1} \right|, \sum_{m=0}^{\lfloor 143/4 \rfloor} A_{m,n}$$

This further implies

$$B_2 = \max \left\{ \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6, \lambda_7, \lambda_8, \lambda_9, \lambda_{10}, \lambda_{11}, \lambda_{12}, \lambda_{13}, \lambda_{14}, \lambda_{15}, \lambda_{16} \right\}$$
  
Since  $b_i = 0$ , for all  $i > 11$ , therefore we have

$$\lambda_1 = \left| b_4 b_{11} + b_8 b_{10} \right| + \left| b_0 b_8 + b_4 b_7 + b_6 b_8 \right| + \left| b_0 b_4 + b_2 b_8 + b_3 b_4 \right| + \left| b_0 \right|^2,$$

$$\lambda_2 = \left| b_5 b_{11} + b_9 b_{10} \right| + \left| b_1 b_8 + b_5 b_7 + b_6 b_9 \right| + \left| b_1 b_4 + b_2 b_9 + b_3 b_5 \right| + \left| b_0 b_1 \right|,$$

$$\lambda_3 = \left| b_6 b_{11} + b_{10}^2 \right| + \left| b_2 b_8 + b_6 b_7 + b_6 b_{10} \right| + \left| b_2 b_4 + b_2 b_{10} + b_3 b_6 \right| + \left| b_0 + b_2 \right|,$$

$$\lambda_4 = \left| b_7 b_{11} + b_{10} b_{11} \right| + \left| b_3 b_8 + b_6 b_{11} + b_7^2 \right| + \left| b_2 b_{11} + b_3 b_4 + b_3 b_7 \right| + \left| b_0 b_3 \right|,$$

$$\lambda_5 = |b_{11}b_8| + |b_0b_9 + b_4b_8 + b_7b_8| + |b_0b_5 + b_3b_8 + b_4^2| + |b_0b_1 + b_0b_4|,$$

$$\lambda_6 = \left| b_{11}b_9 \right| + \left| b_1b_9 + b_5b_8 + b_7b_9 \right| + \left| b_1b_5 + b_3b_9 + b_4b_5 \right| + \left| b_0b_5 + b_1^2 \right|,$$

$$\lambda_7 = \left| b_{10}b_{11} \right| + \left| b_2b_9 + b_6b_8 + b_7b_{10} \right| + \left| b_2b_5 + b_3b_{10} + b_4b_6 \right| + \left| b_0b_6 + b_1b_2 \right|,$$

$$\lambda_{8} = \left| b_{11}^{2} \right| + \left| b_{3}b_{9} + b_{7}b_{8} + b_{7}b_{11} \right| + \left| b_{3}b_{5} + b_{3}b_{11} + b_{4}b_{7} \right| + \left| b_{0}b_{7} + b_{1}b_{3} \right|,$$
  
$$\lambda_{9} = \left| b_{0}b_{10} + b_{4}b_{9} + b_{8}^{2} \right| + \left| b_{0}b_{6} + b_{4}b_{5} + b_{4}b_{8} \right| + \left| b_{0}b_{2} + b_{0}b_{8} + b_{1}b_{4} \right|,$$

$$\lambda_9 = \left| b_0 b_{10} + b_4 b_9 + b_8^2 \right| + \left| b_0 b_6 + b_4 b_5 + b_4 b_8 \right| + \left| b_0 b_2 + b_0 b_8 + b_1 b_4 \right|,$$

$$\lambda_{10} = \left| b_{1}b_{10} + b_{5}b_{9} + b_{8}b_{9} \right| + \left| b_{1}b_{6} + b_{4}b_{9} + b_{5}^{2} \right| + \left| b_{0}b_{9} + b_{1}b_{2} + b_{1}b_{5} \right|,$$

$$\lambda_{11} = \left| b_{2}b_{10} + b_{6}b_{9} + b_{8}b_{10} \right| + \left| b_{2}b_{6} + b_{4}b_{10} + b_{5}b_{6} \right| + \left| b_{0}b_{10} + b_{1}b_{6} + b_{2}^{2} \right|,$$

$$\lambda_{12} = \left| b_{3}b_{10} + b_{7}b_{9} + b_{8}b_{11} \right| + \left| b_{3}b_{6} + b_{4}b_{11} + b_{5}b_{7} \right| + \left| b_{0}b_{11} + b_{1}b_{7} + b_{2}b_{3} \right|,$$

$$\lambda_{13} = \left| b_{0}b_{11} + b_{4}b_{10} + b_{8}b_{9} \right| + \left| b_{0}b_{7} + b_{4}b_{6} + b_{5}b_{8} \right| + \left| b_{0}b_{3} + b_{1}b_{8} + b_{2}b_{4} \right|,$$

$$\lambda_{14} = \left| b_{1}b_{11} + b_{5}b_{10} + b_{9}^{2} \right| + \left| b_{1}b_{7} + b_{5}b_{6} + b_{5}b_{9} \right| + \left| b_{1}b_{3} + b_{1}b_{9} + b_{2}b_{5} \right|,$$

$$\lambda_{15} = \left| b_{2}b_{11} + b_{6}b_{10} + b_{9}b_{10} \right| + \left| b_{2}b_{7} + b_{5}b_{10} + b_{6}^{2} \right| + \left| b_{1}b_{10} + b_{2}b_{3} + b_{2}b_{6} \right|,$$

$$\lambda_{16} = \left| b_{3}b_{11} + b_{7}b_{10} + b_{9}b_{11} \right| + \left| b_{3}b_{7} + b_{5}b_{11} + b_{6}b_{7} \right| + \left| b_{1}b_{11} + b_{2}b_{7} + b_{3}^{2} \right| \right\}.$$

*Now using* (25)*, we acquire* 

$$B_{2} = \max \left\{ |b_{4}b_{11} + b_{8}b_{10}| + |b_{0}b_{8} + b_{4}b_{7} + b_{6}b_{8}| + |b_{0}b_{4} + b_{2}b_{8} + b_{3}b_{4}| + |b_{0}|^{2}, |b_{5}b_{11} + b_{9}b_{10}| + |b_{1}b_{8} + b_{5}b_{7} + b_{6}b_{9}| + |b_{1}b_{4} + b_{2}b_{9} + b_{3}b_{5}| + |b_{0}b_{1}|, |b_{6}b_{11} + b_{1}^{2}_{10}| + |b_{2}b_{8} + b_{6}b_{7} + b_{6}b_{10}| + |b_{2}b_{4} + b_{2}b_{10} + b_{3}b_{6}| + |b_{0} + b_{2}|, |b_{7}b_{11} + b_{10}b_{11}| + |b_{3}b_{8} + b_{6}b_{11} + b_{7}^{2}| + |b_{2}b_{11} + b_{3}b_{4} + b_{3}b_{7}| + |b_{0}b_{3}|, 0 + 0 + |b_{0}b_{5} + b_{3}b_{8} + b_{4}^{2}| + |b_{0}b_{1} + b_{0}b_{4}|, 0 + 0 + |b_{1}b_{5} + b_{3}b_{9} + b_{4}b_{5}| + |b_{0}b_{5} + b_{1}^{2}|, 0 + 0 + |b_{2}b_{5} + b_{3}b_{10} + b_{4}b_{6}| + |b_{0}b_{6} + b_{1}b_{2}|, |b_{1}^{2}_{11}| + |b_{3}b_{9} + b_{7}b_{8} + b_{7}b_{11}| + |b_{3}b_{5} + b_{3}b_{11} + b_{4}b_{7}| + |b_{0}b_{7} + b_{1}b_{3}|, 0 + |b_{0}b_{6} + b_{4}b_{5} + b_{4}b_{8}| + |b_{0}b_{2} + b_{0}b_{8} + b_{1}b_{4}|, 0 + |b_{1}b_{6} + b_{4}b_{9} + b_{5}^{2}| + |b_{0}b_{9} + b_{1}b_{2} + b_{1}b_{5}|, 0 + |b_{2}b_{6} + b_{4}b_{10} + b_{5}b_{6}| + |b_{0}b_{10} + b_{1}b_{6} + b_{2}^{2}, 0 + |b_{3}b_{6} + b_{4}b_{11} + b_{5}b_{7}| + |b_{0}b_{11} + b_{1}b_{7} + b_{2}b_{3}|, |b_{0}b_{11} + b_{4}b_{10} + b_{8}b_{9}| + |b_{0}b_{7} + b_{4}b_{6}| + b_{5}b_{8}| + |b_{0}b_{3} + b_{1}b_{8} + b_{2}b_{4}|, |b_{1}b_{11} + b_{5}b_{10} + b_{6}^{2}| + |b_{1}b_{7} + b_{5}b_{6} + b_{5}b_{9}| + |b_{1}b_{3} + b_{1}b_{9}| + b_{2}b_{7} + b_{5}b_{1} + b_{5}b_{7}| + |b_{1}b_{11} + b_{2}b_{7} + b_{5}^{2}| + |b_{1}b_{10} + b_{2}b_{3} + b_{2}b_{6}|, |b_{3}b_{11} + b_{7}b_{10} + b_{2}b_{1}| + |b_{3}b_{7} + b_{5}b_{11} + b_{6}b_{7}| + |b_{1}b_{11} + b_{2}b_{7} + b_{3}^{2}| \right\}.$$

This implies that

$$B_2 = \max\left\{\frac{1}{16}, \frac{1}{16}, \frac{1}{$$

Similarly, we can compute the values of  $B_{l_0}$  for  $l_0 \ge 3$ . For convenience, we have computed the values up to  $l_0 = 4$ , which are shown in Table 1. The subdivision depth k by Theorem 3 at different values of  $B_{l_0}$  are given in Table 2.

In this work,  $B_{l_0}$  for  $l_0 = 1$  is equal to  $\delta_1$  defined in [15]. If  $\delta_1 > 1$ , then the error bound of QSS cannot be computed. However, in the proposed approach, if we increase the value of  $l_0$ , the values of  $B_{l_0}$  decreases until  $B_{l_0}$  becomes less than one. The main advantage of this approach also to compute error bounds of those QSS, whose  $\delta_1$  is greater or equal to one. As in Table 2, twenty-eight

iterations are needed to achieve a given error tolerance  $8.7 \times 10^{-19}$  by technique given in [15], but by our technique, it needs only seven iterations corresponding to B<sub>4</sub>. The graphical comparison between the results at first and fourth convolutions are demonstrated in Figure 3a.

**Table 1.** Values of  $B_{l_0}$  for  $l_0 = 1, 2, 3, 4$ .

Scheme/B <sub>l0</sub>	$B_1 = \delta_1 [15]$	<i>B</i> <sub>2</sub>	<i>B</i> <sub>3</sub>	$B_4$
3-point approximating curve [2]	0.250000	0.062500	0.015625	0.003906
4-point approximating curve [1]	0.250000	0.062500	0.015625	0.003906
4-point interpolating curve [4]	0.328125	0.105957	0.034286	0.011092

Table 2. Depth of a 3-point approximating QSS curve model.

$B_{l_0}/\epsilon$	$2.45 \times 10^{-3}$	$9.57~ imes~10^{-7}$	$3.74 \times 10^{-9}$	$1.46  imes 10^{-11}$	$5.7 \times 10^{-14}$	$2.23  imes 10^{-16}$	$8.7  imes 10^{-19}$
$B_1 = \delta_1 \ [15]$	4	8	12	16	20	24	28
$B_2$	2	4	6	8	10	12	14
$B_3$	1	3	4	5	7	8	9
$B_4$	1	2	3	4	5	6	7

**Example 2.** The subdivision depth of the curve model produced by a 4-point approximating QSS [1] are given in Table 3. The graphical view of these depths is shown in Figure 3b.

$B_{l_0}/\epsilon$	$4.41 \times 10^{-4}$	$1.72~ imes~10^{-6}$	$6.73 \times 10^{-9}$	$2.63  imes 10^{-11}$	$1.03 imes10^{-13}$	$4.01  imes 10^{-16}$	$1.57 imes10^{-18}$
$B_1 = \delta_1 \ [15]$	4	8	12	16	20	24	28
<i>B</i> <sub>2</sub>	2	4	6	8	10	12	14
$B_3$	1	3	4	5	7	8	9
$B_4$	1	2	3	4	5	6	7

**Table 3.** Depth of a 4-point approximating QSS curve model.

**Example 3.** The subdivision depths of the curve model produced by a 4-point interpolating QSS [4] for  $B_{l_0}, l_0 \ge 1$  (see Table 1) are shown in Table 4. From Table 4, we can observe that the number of iterations k (subdivision depth) decreases with the increase of  $l_0$  (order of convolution) to obtain a user given error tolerance. This is the main reason for the reduction of computational cost as compared to the technique given by [15]. The graphical analysis can be seen in Figure 3c.

Table 4. Depth of a 4-point interpolating QSS curve model.

$B_{l_0}/\epsilon$	$1.12 \times 10^{-3}$	$1.24 \times 10^{-5}$	$1.38 \times 10^{-7}$	$1.53 \times 10^{-9}$	$1.7 \times 10^{-11}$	$1.88 \times 10^{-13}$	$2.09 \times 10^{-15}$
$B_1 = \delta_1 \ [15]$	4	8	12	16	21	25	29
B <sub>2</sub>	2	4	6	8	10	12	14
$B_3$	1	3	4	5	7	8	9
$B_4$	1	2	3	4	5	6	7



Figure 3. The comparison between various convolutions for curve case.

#### 5. Numerical Applications for Surface Models

Now we demonstrate the performance of our results to compute error bound and subdivision depth of the surface models. First, we compute the term  $C_{l_0}D_{l_0}$  for  $l_0 \ge 1$  by using (20) and (21). These are shown in Table 5. We see that the values of  $C_{l_0}D_{l_0}$  decrease with the increase of  $l_0$ . This is the advantage of our approach.

**Table 5.** The values of  $C_{l_0}D_{l_0}$  for  $l_0 = 1, 2, 3, 4$ .

Scheme/ $C_{l_0}D_{l_0}$	$C_1 D_1 = \delta_2 \ [15]$	$C_2D_2$	$C_3D_3$	$C_4D_4$
3-point QSS for surface [2] 4-point QSS for surface [1]	0.250000	0.062500	0.015625	0.003906
4-point QSS for surface [4]	0.389648	0.140310	0.049504	0.017476

**Example 4.** The subdivision depths of the surface model produced by the tensor product of a 3-point approximating QSS [2] are given in Table 6. The values are computed by using the Theorem 6. The graphical representation is shown in Figure 4a.

Table 6. Depth of the 3-point approximating QSS surface model.

$C_{l_0}D_{l_0}/\epsilon$	$7.8 \times 10^{-4}$	$3.05 \times 10^{-6}$	$1.19 \times 10^{-8}$	$4.65~\times~10^{-11}$	$1.82 \times 10^{-13}$	7.1 $ imes$ 10 <sup>-16</sup>	$2.77 \times 10^{-18}$
$C_1 D_1 = \delta 2 [15]$	4	8	12	16	20	24	28
$C_2D_2$	2	4	6	8	10	12	14
$C_3D_3$	1	3	4	5	7	8	9
$C_4D_4$	1	2	3	4	5	6	7

**Example 5.** The subdivision depths of the surface model produced by the tensor product of a 4-point approximating QSS [1] are shown in Table 7. Also, these depths are graphically shown in Figure 4b.

Table 7. Depth of the 4-point approximating QSS surface model.

$C_{l_0}D_{l_0}/\epsilon$	$1.51 \times 10^{-3}$	$6.36 \times 10^{-6}$	$2.68~\times~10^{-8}$	$1.13 \times 10^{-10}$	$4.74 \times 10^{-13}$	1.99 $ imes$ 10 <sup>-15</sup>	$8.4 \times 10^{-18}$
$C_1 D_1 = \delta_2 [15]$	4	8	13	17	21	25	29
$C_2D_2$	2	4	6	8	10	12	14
$C_3D_3$	1	3	4	5	7	8	9
$C_4D_4$	1	2	3	4	5	6	7

**Example 6.** The subdivision depth of a 4-point interpolating QSS surface model [4] corresponding to  $C_{l_0}D_{l_0}$ ,  $l_0 \ge 1$  (see Table 5) is shown in Table 8. The graphical view of the first and fourth convolutions is given in Figure 4c.

Table 8. Depth of a 4-point interpolating QSS surface model.

$C_{l_0}D_{l_0}/\epsilon$	$6.22 \times 10^{-3}$	$1.09  imes 10^{-4}$	$1.9  imes 10^{-6}$	$3.32 \times 10^{-8}$	5.81 $ imes$ 10 <sup>-10</sup>	$1.01 \times 10^{-11}$	$1.77 \times 10^{-13}$
$C_1 D_1 = \delta_2 [15]$	5	9	13	18	22	26	31
$C_2D_2$	2	4	6	8	10	12	14
$C_3D_3$	1	3	4	5	7	8	9
$C_4D_4$	1	2	3	4	5	6	7



Figure 4. The comparison between different convolutions for the surface case.

#### 6. Conclusions

An advance computational technique has been developed to compute the error bounds of the quaternary subdivision model from its control polygon at  $k^{th}$  level. This technique also predict the number of iterations (subdivision depth) which are required to reach user-defined error tolerance. This technique is the modified version of the technique presented in [15]. When the technique of [15] fails to work then the proposed technique can work by increasing the convolution steps. Moreover, we need fewer iterations to get the optimal subdivision depth as compared to the existing techniques.

#### Appendix A

Here, we present the proofs of results given in Section 2 and 3.

## Appendix A.1. Proof of Lemma 1

**Proof.** To prove this result, we start with the case of  $(l_0 = 1)$  and  $(l_0 = 2)$  convolutions then a general case will be derived.

• Case  $l_0 = 1$ 

From (7), we obtain a relation given as follows

$$(p^{(0)} \star b)_j = \sum_{n=0}^{\lfloor j/4 \rfloor} p_n b_{j-4n},$$
 (A1)

where  $\lfloor . \rfloor$  indicates the integer part of '.', this implies

$$|(p^{(0)} \star b)_j| = \left| \sum_{n=0}^{\lfloor j/4 \rfloor} p_n b_{j-4n} \right| \leq ||p||_{\infty} \sum_{n=0}^{\lfloor j/4 \rfloor} |b_{j-4n}|.$$

Then

$$|(p^{(0)} \star b)_j| = max|(p^{(0)} \star b)_j| \leq \max\left(||p||_{\infty} \sum_{n=0}^{\lfloor j/4 \rfloor} |b_{j-4n}|\right) = ||p||_{\infty} \max\left(\sum_{n=0}^{\lfloor j/4 \rfloor} |A_{n,j}^1|\right),$$

where  $b_{j-4n} = A_{n,j}^1$ . Thus

$$\|(p^{(0)} \star b)_j\|_{\infty} \le \|p\|_{\infty} \max\left(\sum_{n=0}^{\lfloor j/4 \rfloor} |A_{n,j}^1|\right).$$
(A2)

• Case  $l_0 = 2$ From (A1), we acquire

$$((p^{(0)} \star b)^{(0)} \star b)_j = \sum_{m=0}^{\lfloor j/4 \rfloor} (p^{(0)} \star b)_m b_{j-4m} = \sum_{m=0}^{\lfloor j/4 \rfloor} \left( \sum_{n=0}^{\lfloor m/4 \rfloor} p_n b_{m-4n} \right) b_{j-4m}.$$

This implies

$$\begin{split} ((p^{(0)} \star b)^{(0)} \star b)_{j} &= p_{0}(b_{0}b_{j} + b_{1}b_{j-4} + b_{2}b_{j-8} + b_{3}b_{j-12} + b_{4}b_{j-16} + \ldots + b_{\lfloor \frac{i}{4} \rfloor}b_{0}) \\ &+ p_{1}(b_{0}b_{j-16} + b_{1}b_{j-20} + \ldots + b_{0}b_{\lfloor \frac{i}{4} \rfloor - 4}) + p_{2}(b_{0}b_{j-32} + b_{1}b_{j-36} + \ldots \\ &+ b_{0}b_{\lfloor \frac{i}{4} \rfloor - 8}) + \ldots + p_{\lfloor \frac{i}{42} \rfloor}b_{0}b_{0} \\ &= p_{0}\left(\sum_{n=0}^{\lfloor j/4 \rfloor} b_{n}b_{j-4n}\right) + p_{1}\left(\sum_{n=4}^{\lfloor j/4 \rfloor} b_{n-4}b_{j-4n}\right) + p_{2}\left(\sum_{n=8}^{\lfloor j/4 \rfloor} b_{n-8}b_{j-4n}\right) + \ldots \\ &+ p_{\frac{i}{42}}\left(\sum_{n=4 \lfloor \frac{i}{42} \rfloor}^{\lfloor j/4 \rfloor} b_{n-4 \lfloor \frac{i}{42} \rfloor}b_{j-4n}\right) \\ &= \sum_{m=0}^{\lfloor j/4^{2} \rfloor} p_{m}\left(\sum_{n=4m}^{\lfloor j/4 \rfloor} b_{n-4m}b_{j-4n}\right) = \sum_{m=0}^{\lfloor j/4^{2} \rfloor} p_{m}\left(\sum_{n=4m}^{\lfloor j/4^{2} \rfloor} p_{m}A_{m,n}^{2}A_{n,j}^{1}\right) = \sum_{m=0}^{\lfloor j/4^{2} \rfloor} p_{m}A_{m,j}^{2}, \end{split}$$

where

$$A_{m,j}^{2} = \sum_{n=4m}^{\lfloor j/4 \rfloor} A_{m,n}^{1} A_{n,j}^{1}.$$
 (A3)

So

$$\left|((p^{(0)}\star b)^{(0)}\star b)_j\right| = \left|\sum_{m=0}^{\lfloor j/4^2 \rfloor} p_m A_{m,j}^2\right|.$$

This implies

$$\|((p^{(0)} \star b)^{(0)} \star b)_j\|_{\infty} \le \|p\|_{\infty} \max_j \left\{ \sum_{m=0}^{\lfloor j/4^2 \rfloor} |A_{m,j}^2| \right\}.$$
(A4)

• The general case

Applying the same argument, we acquire the reformulations for  $l_0^{th}$  convolution as follows

$$((\dots(((p^{(0)} \star b)^{(0)}) \star b)^{(0)} \star \dots \star b)^{(0)} \star b)_j = \sum_{m=0}^{\lfloor j/4^{t_0} \rfloor} p_m A_{m,j}^{l_0},$$

where  $A_{m,j}^{l_0}$  is defined recursively by

$$\begin{cases} A_{m,j}^{1} = b_{j-4m}, \\ A_{m,j}^{l_{0}} = \sum_{p=4m}^{\lfloor j/4^{l_{0}-1} \rfloor} A_{m,p}^{1} A_{p,j}^{l_{0}-1}, \quad l_{0} \ge 2. \end{cases}$$
(A5)

Hence

$$\|((\dots(((p^{(0)} \star b)^{(0)}) \star b)^{(0)} \star \dots \star b)^{(0)} \star b)\|_{\infty} \le \|p\|_{\infty} \max_{j} \left\{ \sum_{m=0}^{\lfloor j/4^{l_{0}} \rfloor} |A_{m,j}^{l_{0}}| \right\}.$$
(A6)

Appendix A.2. Proof of Lemma 2

**Proof.** Here, we start from an induction process over  $l_0$ .

• Case  $l_0 = 1$ 

$$A_{m,j}^{1} = b_{j-4m} = b_{j+4-4(m+1)} = A_{m+1,j+4}^{1},$$
(A7)

16 of 20

similarly

$$A_{m+1,j}^1 = b_{j-4(m+1)} = A_{m,j-4}^1.$$
(A8)

From (A3), we have

$$A_{m,j}^2 = \sum_{n=4m}^{\lfloor j/4 \rfloor} A_{m,n}^1 A_{n,j}^1.$$

Using (A7), we have

$$A_{m,j}^2 = \sum_{n=4m}^{\lfloor j/4 \rfloor} A_{m,n}^1 A_{n+1,j+4j}^1$$

then after replacing *n* by n - 4, we acquire

$$A_{m,j}^2 = \sum_{n=4(m+1)}^{\lfloor j/4+4 \rfloor} A_{m,n-4}^1 A_{n-3,j+4}^1$$

Using (A8)

$$A_{m,j}^2 = \sum_{n=4(m+1)}^{\lfloor j/4+4 \rfloor} A_{m+1,n}^1 A_{n,j+4^2}^1 = A_{m+1,j+4^2}^2$$

We suppose that it is true for an integer  $l_0 = M$  that is

$$A_{m,j}^{M} = A_{m+1,j+4^{M}}^{M}.$$
 (A9)

Now, we will prove the statement for

• Case  $l_0 = M + 1$ Consider

$$A_{m,j}^{M+1} = \sum_{n=4m}^{\lfloor j/4^M \rfloor} A_{m,n}^1 A_{n,j}^M$$

By using (A9), we acquire

$$A_{m,j}^{M+1} = \sum_{n=4m}^{\lfloor j/4^M \rfloor} A_{m,n}^1 A_{n+1,j+4^M}^M.$$

Now, replace *n* by n - 4

$$A_{m,j}^{M+1} = \sum_{n=4(m+1)}^{\lfloor j/4^M + 4 \rfloor} A_{m,n-4}^1 A_{n-3,j+4^M}^M.$$

Using (A8) and (A9), we have

$$A_{m,j}^{M+1} = A_{m+1,j+4^{M+1}}^{M+1}.$$

Similarly we can prove

$$A_{m,j}^{M+1} = A_{m-1,j-4^{M+1}}^{M+1}.$$

Hence

$$A_{m-1,j-4^{l_0}}^{l_0} = A_{m,j}^{l_0} = A_{m+1,j+4^{l_0}}^{l_0}.$$

This completes the proof.  $\Box$ 

Appendix A.3. Proof of Corollary 1

**Proof.** Assume that  $b = \{b_0, b_1, \dots, b_{4N-1}\}$ , with  $N \in \mathbb{N}$  and  $\Omega(l_0, N) = (4^{l_0} - 3)(4N - 1)$ . Then for  $j > \Omega(l_0, N)$  and by using Lemma 1, we acquire

$$A_{0,i}^{l_0} = 0. (A10)$$

Similarly for  $j > \Omega(l_0, N) + m4^{l_0}$  and using Lemma 2, we have

$$A_{m,i}^{l_0} = 0. (A11)$$

Finally, using (A10) and (A11), we get (13).  $\Box$ 

## Appendix A.4. Proof of Lemma 3

**Proof.** To prove the proposed result, we start with the case of  $(l_0 = 1)$ th and  $(l_0 = 2)$ th convolutions then we discuss the general case.

• Case  $l_0 = 1$ 

Consider an arbitrary sequence of vectors  $p_{i,j}$ . Then we have

$$p_{i,j}^{1} = (p^{0;0} \star cd)_{i,j} = \sum_{m=0}^{\lfloor i/4 \rfloor} \sum_{n=0}^{\lfloor j/4 \rfloor} p_{m,n}^{0} c_{i-4m} d_{j-4n}$$

where we are taking  $A_{m,i}^{1,c} = c_{i-4m}$  and  $d_{j-4n} = A_{n,j}^{1,d}$  for *c* and *d*. Thus

$$p_{i,j}^{1} = (p^{0;0} \star cd)_{i,j} = \sum_{m=0}^{\lfloor i/4 \rfloor} \sum_{n=0}^{\lfloor j/4 \rfloor} p_{m,n}^{0} A_{m,i}^{1,c} A_{n,j}^{1,d}$$

This implies

$$\max_{i,j} |p_{i,j}^{1}| = \max_{i,j} \left| \sum_{m=0}^{\lfloor i/4 \rfloor} \sum_{n=0}^{\lfloor j/4 \rfloor} p_{m,n}^{0} A_{m,i}^{1,c} A_{n,j}^{1,d} \right|, \\
\leq \max_{i,j} \sum_{m=0}^{\lfloor i/4 \rfloor} \sum_{n=0}^{\lfloor j/4 \rfloor} |A_{m,i}^{1,c}| |A_{n,j}^{1,d}| \max_{m,n} |p_{m,n}^{0}|.$$
(A12)

Consider

and

$$C_1 = \max_i \left\{ \sum_{m=0}^{\lfloor i/4 \rfloor} |A_{m,i}^{1,c}| \right\}$$
$$D_1 = \max_j \left\{ \sum_{n=0}^{\lfloor j/4 \rfloor} |A_{n,j}^{1,d}| \right\},$$

then from (A12), we obtain

 $\max_{i,j} |p_{i,j}^1| \le C_1 D_1 \max_{m,n} |p_{m,n}^0|.$ 

• Case  $l_0 = 2$ 

Now, after applying two time convolution, we obtain

$$p_{m,n}^{1} = (p^{0;0} \star cd)_{m,n} = ((p^{1;0} \star cd) \star cd)_{i,j}$$

This implies

$$p_{i,j}^{1} = \sum_{m=0}^{\lfloor i/4 \rfloor} \sum_{n=0}^{\lfloor j/4 \rfloor} (p^{1;0} \star cd)_{i,j} c_{i-4m} d_{j-4n}$$
$$= \sum_{m=0}^{\lfloor i/4 \rfloor} \sum_{n=0}^{\lfloor j/4 \rfloor} \left( \sum_{p=0}^{\lfloor m/4 \rfloor} \sum_{s=0}^{\lfloor n/4 \rfloor} p_{p,s}^{0} c_{m-4p} d_{n-4s} \right) c_{i-4m} d_{j-4n}$$

This again implies that

$$p_{i,j}^{1} = \sum_{m=0}^{\lfloor i/4^{2} \rfloor} \sum_{n=0}^{\lfloor j/4^{2} \rfloor} p_{m,n}^{0} \sum_{r=4m}^{\lfloor i/4 \rfloor} c_{r-4m} c_{i-4r} \sum_{q=4n}^{\lfloor j/4 \rfloor} d_{q-4n} d_{j-4q}$$
$$= \sum_{m=0}^{\lfloor i/4^{2} \rfloor} \sum_{n=0}^{\lfloor j/4^{2} \rfloor} p_{m,n}^{0} \sum_{r=4m}^{\lfloor i/4 \rfloor} A_{m,r}^{1,c} A_{r,i}^{1,c} \sum_{q=4n}^{\lfloor j/4 \rfloor} A_{n,q}^{1,d} A_{q,j}^{1,d}.$$

Which implies

$$p_{i,j}^2 = \sum_{m=0}^{\lfloor i/4^2 \rfloor} \sum_{n=0}^{\lfloor j/4^2 \rfloor} p_{m,n}^0 A_{m,i}^{2,c} A_{n,j}^{2,d}.$$

Now

$$\max_{i,j} |p_{i,j}^{2}| = \max_{i,j} \left| \sum_{m=0}^{\lfloor i/4^{2} \rfloor} \sum_{n=0}^{\lfloor j/4^{2} \rfloor} p_{m,n}^{0} A_{m,i}^{2,c} A_{n,j}^{2,d} \right|, \\
\leq \max_{i,j} \sum_{m=0}^{\lfloor i/4^{2} \rfloor} \sum_{n=0}^{\lfloor j/4^{2} \rfloor} \left| A_{m,i}^{2,c} \right| \left| A_{n,j}^{2,d} \right| \max_{m,n} \left| p_{m,n}^{0} \right|.$$
(A13)

Consider

and

then we get

$$\max_{i,j} |p_{i,j}^2| \le C_2 D_2 \max_{m,n} |p_{m,n}^0|.$$

 $C_{2} = \max_{i} \left\{ \sum_{m=0}^{\lfloor i/4^{2} \rfloor} \left| A_{m,i}^{2,c} \right| \right\}$ 

 $D_2 = \max_{j} \left\{ \sum_{n=0}^{\lfloor j/4^2 \rfloor} \left| A_{n,j}^{2,d} \right| \right\},$ 

• The general case

By the same strategy, we acquire reformulations for  $(l_0)$ th convolutions given in the following

$$p_{i,j}^{l_0} = (p^{l_0 - l_0; 0} \star cd)_{m,n} = (\dots (((p^{l_0 - 1; 0} \star cd) \star cd) \star \dots \star cd) \star cd)_{i,j}.$$

This implies

$$p_{i,j}^{l_0} = \sum_{m=0}^{\lfloor i/4^{l_0} \rfloor} \sum_{n=0}^{\lfloor j/4^{l_0} \rfloor} p_{m,n}^{0;0} A_{m,i}^{l_0,c} A_{n,j}^{l_0,d} = \sum_{m=0}^{\lfloor i/4^{l_0} \rfloor} \sum_{n=0}^{\lfloor j/4^{l_0} \rfloor} p_{m,n}^0 A_{m,i}^{l_0,c} A_{n,j}^{l_0,d},$$

where

$$A_{m,i}^{l_0,c} = \sum_{p=4m}^{\lfloor i/4^l_0 - 1 \rfloor} A_{m,p}^{1,c} A_{p,i}^{l_0 - 1,c}$$

and

Thus

$$\max_{i,j} |p_{i,j}^{l_0}| = \max_{i,j} \left| \sum_{m=0}^{\lfloor i/4^{l_0} \rfloor} \sum_{n=0}^{\lfloor j/4^{l_0} \rfloor} p_{m,n}^0 A_{m,i}^{l_0,c} A_{n,j}^{l_0,d} \right|, \\
\leq \max_{i,j} \sum_{m=0}^{\lfloor i/4^{l_0} \rfloor} \sum_{n=0}^{\lfloor j/4^{l_0} \rfloor} \left| A_{m,i}^{l_0,c} \right| \left| A_{n,j}^{l_0,d} \right| \max_{m,n} \left| p_{m,n}^0 \right|.$$
(A14)

Now consider

and

 $D_{l_0} = \max_j \left\{ \sum_{n=0}^{\lfloor j/4^{l_0} 
floor} |A_{n,j}^{l_0,d}| 
ight\},$ 

 $C_{l_0} = \max_{i} \left\{ \sum_{m=0}^{\lfloor i/4^{l_0} \rfloor} |A_{m,i}^{l_0,c}| \right\}$ 

 $A_{n,j}^{l_{0},d} = \sum_{q=4n}^{\lfloor j/4^{l_{0}}-1 \rfloor} A_{n,q}^{1,d} A_{q,j}^{l_{0}-1,d}.$ 

then, from (A14), we obtain

 $\max_{i,j} |p_{i,j}^{l_0}| \le C_{l_0} D_{l_0} \max_{m,n} |p_{m,n}^0|,$ 

where

$$\max_{i,j} \left\{ \sum_{m=0}^{\lfloor i/4^{l_0} \rfloor} \sum_{n=0}^{\lfloor j/4^{l_0} \rfloor} |A_{n,j}^{l_0,d}| |A_{m,i}^{l_0,c}| \right\} = \max_{i,j \in \Sigma(l_0,N)} \left\{ \sum_{m=0}^{\lfloor i/4^{l_0} \rfloor} \sum_{n=0}^{\lfloor j/4^{l_0} \rfloor} |A_{n,j}^{l_0,d}| |A_{m,i}^{l_0,c}| \right\}.$$
(A15)

**Author Contributions:** Conceptualization, F.K. and S.-W.Y.; Formal analysis, M.I. and S.A.; Methodology, A.G. and S.-W.Y.; Supervision, A.G.; Writing—original draft, A.S., F.K. and M.I.; Writing—review & editing, A.G. and F.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** National Natural Science Foundation of China (No. 71601072), Key Scientific Research Project of Higher Education Institutions in Henan Province of China (No. 20B110006) and the Fundamental Research Funds for the Universities of Henan Province.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data for implementation of the result are included in the paper.

Acknowledgments: The author A. Shahzad thanks to University of Sargodha, Pakistan for providing facilities and support.

Conflicts of Interest: The authors declare no conflict of interest in this paper.

#### References

- Mustafa, G.; Faheem, K. A new 4-point C<sup>3</sup> quaternary approximating subdivision scheme. *Abstr. Appl. Anal.* 2009, *9*, 1–14. [CrossRef]
- Siddiqi, S.S.; Younis, M. The m-point quaternary approximating subdivision schemes. Am. J. Comput. Math. 2013, 13, 6–10. [CrossRef]
- 3. Siddiqi, S.S.; Younis, M. The quaternary interpolating scheme for geometric design. ISRN Comput. Graph. 2013, 13, 1–8. [CrossRef]
- 4. Pervaiz, K. Shape preservation of the stationary 4-point quaternary subdivision schemes. Commun. Math. Appl. 2018, 9, 249–264.
- 5. Deslauriers, G.; Dubuc, S. Symmetric iterative interpolation processes. In *Constructive Approximation*; DeVore, R.A., Saff, E.B., Eds.; Springer: Boston, MA, USA, 1989; pp. 49–68.

- Zheng, H.C.; Song, Q. Designing general p-ary n-point smooth subdivision schemes. *Appl. Mech. Mater.* 2014, 472, 510–515. [CrossRef]
- Conti, C.; Romani, L. Dual univariate a-ary subdivision schemes of de Rham-type. J. Math. Anal. Appl. 2013, 407, 443–456. [CrossRef]
- 8. Lian, J.-A. On *a*-ary subdivision for curve design III. 2m-point and (2m + 1)-point interpolatory schemes. *Appl. Appl. Math. Int. J.* **2009**, *4*, 434–444
- 9. Siddiqi, S.S.; Ahmad, N. A new five-point approximating subdivision scheme. Int. J. Comput. Math. 2008, 85, 65–72. [CrossRef]
- Pan, J.; Lin, S.; Luo, X. A combined approximating and interpolating subdivision scheme with C<sup>2</sup> continuity. *Appl. Math. Lett.* 2012, 25, 2140–2146. [CrossRef]
- 11. Hussain, S.M.; Aziz, U.R.; Baleanu, D.; Nisar, K.S.; Ghaffar, A.; Karim, S.A.A. Generalized 5-point Approximating Subdivision Scheme of Varying Arity. *Mathematics* 2020, *8*, 1–25. [CrossRef]
- 12. Aslam, M.; Abeysinghe, W.P. Odd-ary approximating subdivision schemes and RS strategy for irregular dense initial data. *ISRN Math. Anal.* 2012, 2012, 745096. [CrossRef]
- 13. Deng, C.; Li, Y.; Xu, H. Repeated local operations for m-ary 2N-point Dubuc–Deslauriers subdivision schemes. *Comput. Aided Geom. Des.* **2016**, *44*, 10–14. [CrossRef]
- 14. Mustafa, G.; Chen, F.L.; Deng, J.S. Estimating error bounds for binary subdivision curves/surfaces. *J. Comput. Appl. Math.* 2006, 193, 596–613. [CrossRef]
- 15. Mustafa, G.; Hashmi, S. Estimating error bounds for quaternary subdivision schemes. J. Math. Anal. Appl. 2009, 10 159–167.
- 16. Mustafa, G.; Hashmi, S. Subdivision depth computation for *n*-ary subdivision curves/surfaces. *Vis. Comput.* **2010**, *26*, 841–851. [CrossRef]
- 17. Deng, C.; Jin, W.; Li, Y.; Xu, H. A formula for estimating the deviation of a binary interpolatory subdivision curve from its data polygon. *Appl. Math. Comput.* **2017**, 304, 10–19. [CrossRef]
- 18. 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. [CrossRef]
- 19. 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*, 1–13. [CrossRef]
- 20. Khan, F.; Mustafa, G.; Shahzad, A.; Baleanu, D.; Al-Qurashi, M.M. A computational method for subdivision depth of ternary schemes. *Mathematics* 2020, *8*, 1–22. [CrossRef]