Generalization of Binary Tensor Product Schemes Depends upon Four Parameters

ROBINA BASHIR*, MEHWISH BARI*, AND GHULAM MUSTAFA*

ABSTRACT

This article deals with general formulae of parametric and non parametric bivariate subdivision scheme with four parameters. By assigning specific values to those parameters we get some special cases of existing tensor product schemes as well as a new proposed scheme. The behavior of schemes produced by the general formulae is interpolating, approximating and relaxed. Approximating bivariate subdivision schemes produce some other surfaces as compared to interpolating bivariate subdivision schemes. Polynomial reproduction and polynomial generation are desirable properties of subdivision schemes. Capability of polynomial reproduction and polynomial generation is strongly connected with smoothness, sum rules, convergence and approximation order. We also calculate the polynomial generation and polynomial reproduction of 9-point bivariate approximating subdivision scheme. Comparison of polynomial reproduction, polynomial generation and continuity of existing and proposed schemes has also been established. Some numerical examples are also presented to show the behavior of bivariate schemes.

Key Words: Parametric Bivariate Schemes, Continuity, Polynomial Reproduction, Polynomial Generation.

1.   INTRODUCTION

Subdivision is an efficient tool in geometric modeling. Recently, a lot of work has been done on subdivision scheme in different fields such as Computer Graphics, Computer Simulation and Computer Aided Geometric Design. One can get a smooth curve/surface after applying the subdivision scheme on an initial control polygon/mesh. Surface modeling is a powerful tool for computer graphics and shape presentation. The common approaches in subdivision scheme is interpolating and approximating subdivision. Approximating schemes generate smooth curves/surfaces without preserving the initial shape of control polygon/ mesh while interpolating schemes are used to smooth the limit curves/surfaces by preserving the original/initial shape of control polygon/mesh.

First of all, Rham [1] introduced approximating subdivision scheme and later on, Chaikin [2] proposed the corners cutting approximating subdivision scheme. Initially, approximating schemes were introduced. Further, Dubuc [3] introduced the interpolating 4-point subdivision scheme. The first parametric interpolating

Corresponding Author (E-Mail: roba_bashir@hotmail.com)
* Department of Mathematics, The Islamia University, Bahawalpur.

Mehran University Research Journal of Engineering & Technology, Volume 37, No. 1, January, 2018 [p-ISSN: 0254-7821, e-ISSN: 2413-7219]

119

4-point binary subdivision scheme which generates C^l curves is introduced by Dyn et.al. [4]. Zhenget. al. [5] presented a family of binary subdivision schemes which depends upon several parameters. This family of schemes generates the curves with high continuity. Mustafa et. al. [6] proposed the *m*-point binary parametric approximating subdivision scheme. Shen and Huang [7] introduced a family of curve subdivision schemes with multi-parameters and by setting appropriate parameters some classical curve subdivision schemes can be obtained. Mustafa et. al. [8] presented a family of binary univariate dual and primal subdivision schemes.

Subdivision schemes are computationally competent algorithms for representing smooth surfaces by applying a few steps of a refinement operators. In case of quadrilateral meshes, the new mesh from the old one is generated by a topological and geometrical rules. Most of the well-known surfaces subdivision schemes, which generalized the tensor product of bicubic and biquadratic B-splines are introduced by Catmull and Clark [9], Doo and Sabin [10] respectively. Lane and Riesenfeld [11] provides a framework to generate the uniform B-spline curves and their tensor product extensions by subdivision process. Dyn [12] introduced a new butterfly interpolating subdivision scheme for surface modeling with a shape parameter that provides flexibility of design. Ghaffaret. al. [13] used a unified technique to design tensor product scheme. Kobbelt [14] and Zorinet. al. [15] introduced generalized form of surface modeling of univariate schemes presented by [4]. Mustafa and Randhawa [16] presented a 3-point parametric approximating subdivision scheme and its tensor product version for the generation of regular surfaces with its analysis. Mustafa et. al. [17] introduced generalized and unified families of *p*-ary (2*n*) and (2*n*-1)-point interpolating subdivision schemes of lower and higher arity originated from Lagrange

polynomial and also presented tensor product version of these families of schemes.

In this paper, we are going to generate interpolating, approximating and relaxed subdivision schemes with and without shape parameter for regular surface modeling on quad meshes. By suitable value of tension parameter, the shapes of the limit surfaces can easily controlled. By the general formula, we can also design a new non-symmetric bivariate subdivision scheme. Section 2 gives the general approach of tensor product subdivision schemes. In Section 3, we have calculated the polynomial generation and reproduction of bivariate schemes. In this section, we also establish a comparison table based on continuity, polynomial generation and polynomial reproduction of existing and proposed schemes. Section 4 is based on numerical examples of existing and proposed tensor product schemes. Finally, conclusion is presented in Section 5.

2. ALGORITMS FOR TENSOR PRODUCT SCHEMES

We are going to construct the general formulas for bivariate approximating, interpolating and relaxed subdivision schemes.

2.1 Univariate Schemes

General formula of univariate binary subdivision scheme [5] is:

$$
\begin{cases}
f_{2i}^{k+1} = \displaystyle\sum_{q=0}^{l+1} \alpha_{2q+1} f_{i+q}^{k} \\
f_{2i+1}^{k+1} = \displaystyle\sum_{q=0}^{l+2} \alpha_{2q} f_{i+q}^{k}
\end{cases}
\tag{1}
$$

where $i, k \in Z$, and $l \in Z^{+*}$ where i shows the notation of new points, k is the subdivision level and l is positive integer.

Mehran University Research Journal of Engineering & Technology, Volume 37, No. 1, January, 2018 [p-ISSN: 0254-7821, e-ISSN: 2413-7219]

120

$$\begin{cases} \alpha_0 = aC_{2l+1}^0 \\ \alpha_1 = aC_{2l+1}^1 + bC_{2l+1}^0 \\ \alpha_2 = aC_{2l+1}^2 + bC_{2l+1}^1 + cC_{2l+1}^0 \\ \alpha_3 = aC_{2l+1}^3 + bC_{2l+1}^2 + cC_{2l+1}^1 + dC_{2l+1}^0 \\ \vdots \\ \alpha_{2l} = aC_{2l+1}^{2l} + bC_{2l+1}^{2l-1} + cC_{2l+1}^{2l-2} + dC_{2l+1}^{2l-3} \\ \alpha_{2l+1} = aC_{2l+1}^{2l} + bC_{2l+1}^{2l} + cC_{2l+1}^{2l-1} + dC_{2l+1}^{2l-2} \\ \alpha_{2l+2} = aC_{2l+1}^{2l+1} + bC_{2l+1}^{2l} + dC_{2l+1}^{2l-1} \\ \alpha_{2l+3} = aC_{2l+1}^{2l+1} + dC_{2l+1}^{2l} \\ \alpha_{2l+4} = d \end{cases} \qquad (2)$$

a,b,c,d are parameters, $a+b+c+d = 1/d^t$, $t=2l$ where $t \in Z+$, $l = 1,2,3,...$ and

$$C_{2l+1}^r = \frac{(2l+1)!}{r!(2l+1-r)!}$$

for $r = 0,1,2,...2l+1$

2.2 Bivariate Schemes

By changing the notation "i by j, q by p" in Equation (1) and using the tensor product procedure, we can derive the general formula of bivariate schemes

$$\begin{cases} f_{2i,2j}^{k+1} = \sum_{q=0}^{l+1} \sum_{p=0}^{l+1} \alpha_{2q+1}\alpha_{2p+1} f_{i+q,j+p}^k \\ f_{2i,2j+1}^{k+1} = \sum_{q=0}^{l+1} \sum_{p=0}^{l+2} \alpha_{2q+1}\alpha_{2p} f_{i+q,j+p}^k \\ f_{2i+1,2j}^{k+1} = \sum_{q=0}^{l+2} \sum_{p=0}^{l+1} \alpha_{2q}\alpha_{2p+1} f_{i+q,j+p}^k \\ f_{2i+1,2j+1}^{k+1} = \sum_{q=0}^{l+2} \sum_{p=0}^{l+2} \alpha_{2q}\alpha_{2p} f_{i+q,j+p}^k \end{cases} \qquad (3)$$

The details to get the values of the coefficients presented in Equation (3) can be check by [5] which we used to solve the Equation (2).

By substituting $l = 1$, we get general tensor product scheme as follows:

$$f_{2i,2j}^{k+1} = (2a+b)^2 f_{i,j}^k + (2a+b)(2a+2c+d) f_{i,j+1}^k + (2a+b) df_{i,j+2}^k + (b+2c+d)(2a+b)$$
$$f_{i+1,j}^k + (b+2c+d)^2 f_{i+1,j+1}^k + (b+2c+d) df_{i+1,j+2}^k + (2a+b) df_{i+2,j}^k + d(b+2c+d) f_{i+2,j+1}^k + d^2 f_{i+2,j+2}^k$$
$$f_{2i,2j}^{k+1} = a(2a+b) f_{i,j}^k + a(b+2c+d) f_{i,j+1}^k adf_{i,j+2}^k + (b+2c+d)(2a+b) f_{i+1,j}^k + (2a+2b+c)(b+2c+d)$$
$$f_{i+1,j}^k + (a+2b+c) df_{i+1,j+2}^k + (c+2d)(2a+b) f_{i+2,j}^k + (c+2d)(b+2c+d) f_{i+2,j+1}^k + (c+2d) df_{i+2,j+2}^k$$
$$f_{2i,2j+1}^{k+1} = a(2a+b) f_{i,j}^k + (2a+b)(a+2b+c) f_{i,j+1}^k (2a+b)(c+2d) f_{i,j+2}^k + (b+2c+d)$$
$$af_{i+1,j}^k + (a+2b+c)(b+2c+d) f_{i+1,j+1}^k + (b+2c+d)(c+2d) f_{i+1,j+2}^k + d(a+2b+c) f_{i+2,j+1}^k + (c+2d) df_{i+2,j+2}^k \qquad (4)$$
$$f_{2i+1,2j+1}^{k+1} = a^2 f_{i,j}^k + a(a+2b+c) f_{i,j+1}^k + a(c+2d) f_{i,j+2}^k + (a+2b+c) af_{i+1,j}^k + (a+2b+c)^2$$
$$f_{i+1,j+1}^k + (a+2b+c)(c+2d) f_{i+1,j+2}^k + (c+2d) df_{i+2,j}^k + (c+2d)(a+2b+c) f_{i+2,j+1}^k + (c+2d)^2 df_{i+2,j+2}^k$$

- By letting $a = -\frac{1}{12}$, $b = \frac{1}{4}$, $c = \frac{1}{8}$ and $d = -\frac{1}{24}$ in Equation (4). We get a new 16-point tensor product interpolating scheme.

- By substituting $a = d = \frac{1}{32}$ and $b = c = \frac{3}{32}$ in Equation (4). We get 16-point relaxed tensor product approximating scheme of [5].

- By taking $a = -\frac{3}{32} + \mu$, $b = \frac{7}{17} - 2\mu$, $c = -\frac{3}{32} + \mu$ and $d = 0$ in Equation (4). We get 9-point tensor product approximating scheme of [18].

By a slight variation on the scheme presented in Equation (1), such that by replacing "$l+1$ by $l+2$" in first equation of Equation (1) and adopting the same procedure of tensor product approach, we get another general formula of bivariate schemes mentioned below:

$$\begin{cases} f_{2i,2j}^{k+1} = \sum_{q=0}^{l+2} \sum_{p=0}^{l+2} \alpha_{2q+1}\alpha_{2p+1} f_{i+q,j+p}^k \\ f_{2i,2j+1}^{k+1} = \sum_{q=0}^{l+2} \sum_{p=0}^{l+2} \alpha_{2q+1}\alpha_{2p} f_{i+q,j+p}^k \\ f_{2i+1,2j}^{k+1} = \sum_{q=0}^{l+2} \sum_{p=0}^{l+2} \alpha_{2q}\alpha_{2p+1} f_{i+q,j+p}^k \\ f_{2i+1,2j+1}^{k+1} = \sum_{q=0}^{l+2} \sum_{p=0}^{l+2} \alpha_{2q}\alpha_{2p} f_{i+q,j+p}^k \end{cases} \qquad (5)$$

Mehran University Research Journal of Engineering & Technology, Volume 37, No. 1, January, 2018 [p-ISSN: 0254-7821, e-ISSN: 2413-7219]

121

where the coefficients in Equation (5) can be calculated by replacing "l by $l+\dfrac{1}{2}$" in Equation (3), we get a system of equations with free parameters a, b, c, d such that $a+b+c+d = \dfrac{1}{2^l}, t = 2l+1$

where

$t \in Z^+, l = 0,1,2,...$

and

$$C_{2l+2}^r = \dfrac{(2l+2)!}{r!(2l+2-r)!}$$

for $r = 0,1,2,\dots,2l+2$.

By substituting $l=0$, we get general tensor product scheme as follows:

$$
\begin{aligned}
f_{2i,2j}^{k+1} &= (3a+b)^2 f_{i,j}^k + (3a+b)(a+3b+3c+d)f_{i,j+1}^k + (3a+b)(c+3d)f_{i,j+2}^k + (a+3b+3c+d)(3a+b)\\
f_{i+1,j}^k &+ (a+3b+3c+d)^2 f_{i+1,j+1}^k + (a+3b+3c+d)(c+3d)f_{i+1,j+2}^k + (3a+b)(c+3d)f_{i+2,j}^k + (c+3d)(a+3b)\\
f_{i+2,j+1}^k &+ (c+3d)^2 f_{i+2,j+2}^k\\
f_{2i+1,2j}^{k+1} &= a(3a+b)f_{i,j}^k + a(a+3b+3c+d)f_{i,j+1}^k + a(c+3d)f_{i,j+2}^k + (3a+3b+c)(3a+b)\\
f_{i+1,j}^k &+ (3a+3b+c)(a+3b+3c+d)f_{i+1,j+1}^k + (3a+3b+c)(c+3d)f_{i+1,j+2}^k + (b+3c+3d)(3a+b)\\
f_{i+2,j}^k &+ (b+3b+c)(a+3b+3c+d)f_{i+2,j+1}^k + a(b+3c+3d)(c+3d)f_{i+2,j+2}^k + d(3a+b)\\
f_{i+3,j}^k &+ d(a+3b+3c+d)f_{i+3,j+1}^k + d(c+3d)f_{i+3,j+2}^k\\
f_{2i+2,j+1}^{k+1} &= a(3a+b)f_{i,j}^k + (3a+b)(3a+3b+c)f_{i,j+1}^k + (3a+b)(b+3c+3d)f_{i,j+2}^k + d(3a+b)\\
f_{i+1,j}^k &+ a(a+3b+3c+d)f_{i+1,j+1}^k + (3a+3b+c)(a+3b+d)f_{i+1,j+1}^k + (a+3b+3c+d)(b+3c+3d)\\
f_{i+2,j}^k &+ (a+3b+3c+d)f_{i+2,j+1}^k + a(c+3d)f_{i+2,j+2}^k + (c+3d)(3a+3b+c)f_{i+3,j}^k + (b+3c+3d)(c+3d)\\
f_{i+3,j+1}^k &+ d(c+3d)f_{i+3,j+2}^k\\
f_{2i+1,j+1}^{k+1} &= a^2 f_{i,j}^k + a(3a+3b+c)f_{i,j+1}^k + a(b+3c+3d)f_{i,j+2}^k + daf_{i,j+3}^k + (3a+3b+c)\\
f_{i,j+1}^k &+ (3a+3b+c)^2(3a+3b+c)f_{i+1,j+1}^k + (3a+3b+c)(b+3c+3d)f_{i+2,j+2}^k + (3a+3b+c)df_{i+1,j+3}^k(b+3c+d)\\
af_{i+2,j}^k &+ (b+3c+3d)(3a+3b+c)f_{i+2,j+1}^k(b+3c+3d)^2 f_{i+2,j+2}^k + (b+3c+3d)df_{i+2,j+3}^k + daf_{i+3,j}^k + d(3a+3b+c)\\
f_{i+3,j+1}^k &+ d(b+3c+3d)f_{i+3,j+2}^k + d^2 f_{i+3,j+3}^k
\end{aligned}
\tag{6}
$$

- By substituting $a = b = 0$ and $c = d = \dfrac{1}{4}$ in Equation (6) we get 4-point tensor product approximating scheme of [2].

- By substituting $a = c = \dfrac{1}{8}$ and $b = \dfrac{1}{4}, d = 0$ in Equation (6) we get 9-point tensor product approximating scheme of [5].

- By substituting $a = \dfrac{1}{8}$ b $= \dfrac{1}{4}$ C $= \dfrac{3}{8}$ and $d = \dfrac{1}{4}$ in Equation (6) we get 9-point tensor product interpolating scheme of [7].

- By substituting $a = d = \dfrac{1}{16}$ and $b = c = \dfrac{3}{16}$ in Equation (6) we get 9-point tensor product approximating scheme of [19].

- By substituting $a = d = -\dfrac{3}{32}$ and $b = \dfrac{11}{32}$ in Equation (6) we get 9-point tensor product approximating scheme of [20].

- By substituting $a = d = -\dfrac{1}{32}$ and $b = c = \dfrac{7}{32}$ in Equation (6) we get 9-point tensor product approximating scheme of [21].

- By taking $a = d = \dfrac{\mu}{16}$ and $b = c\dfrac{4-\mu}{16}$ in Equation (6) we get 9-point tensor product approximating scheme of [22].

- By taking $a = d = -3\omega$ and $b = c = \dfrac{1}{4} + 3\omega$ in Equation (6) we get 9-point tensor product approximating scheme of [23].

By substituting $l = 1$ in Equation (5) we get another general tensor product scheme and after substituting different values of parameter, different existing schemes becomes the special case of general tensor product scheme:

- If we set $a = c = -\dfrac{1}{16}, b = \dfrac{1}{4}$ and $d = 0$ in Equation (5) we get 16-point tensor product interpolating scheme of [4].

- By taking $a = d = \dfrac{1+6\omega}{96}$ and $b = c = \dfrac{5-6\omega}{96}$ in Equation (5) we get 16-point tensor product approximating scheme of [6].

- Substituting $a = d = \dfrac{1}{64}$ and $b = c = \dfrac{3}{64}$ in Equation (5) we get 16-point tensor product approximating scheme of [19].

- For $a = d = \dfrac{\omega}{16}$, and $b = c = \dfrac{1-\omega}{16}$ in Equation (5) we get 16-point tensor product approximating scheme of [19].

- By putting $a = d = -\dfrac{5}{128}$ and $b = c = \dfrac{13}{128}$ in Equation (5) we get 16-point tensor product approximating scheme of [21].

- By setting $a = d = \frac{1}{384}$ and $b = c = \frac{23}{384}$ in Equation (5) we get 16-point tensor product approximating scheme of [24].

- If we substitute $a = d = \frac{u_o}{64}$ and $b = c = \frac{4 - u_o}{64}$ in Equation (5) we get 16-point tensor product approximating scheme of [25].

3. POLYNOMIAL GENERATION AND REPRODUCTION OF BIVARIATE SCHEMES

In this section, we will investigate the capability of the tensor product approximating subdivision scheme by polynomial generation and polynomial reproduction.

The Laurent polynomial of the scheme Equation (6) for $a = d = \frac{1}{16}$ and $b = c = \frac{3}{16}$ is given by:

$$L(Z_1, Z_2) = \frac{1}{256}(1 + Z_1)^5 (1 + Z_2)^5$$

Theorem-1: If $a = d = \frac{1}{16}$ and $b = c = \frac{3}{16}$, then the subdivision scheme Equation (6) generates polynomial of degree 4.

Proof: Let $\vartheta_1 = (1, -1)$, $\vartheta_2 = (-1,1)$, $\vartheta_3 = (-1,-1)$, and let D^j with $j \in N^2$, denote a directional

derivative. Since $L(1,1) = 4$ and

$D^{(1,0)}L(\vartheta_1) = 0, D^{(1,0)}L(\vartheta_2) = 0, D^{(1,0)}L(\vartheta_3) = 0,$

$D^{(0,1)}L(\vartheta_1) = 0, D^{(01,)}L(\vartheta_2) = 0, D^{(0,1)}L(\vartheta_3) = 0,$

then scheme (6) generates polynomial of degree 1. Again since

$D^{(1,1)}L(\vartheta_1) = 0, D^{(1,1)}L(\vartheta_2) = 0, D^{(1,1)}L(\vartheta_3) = 0,$

$D^{(2,0)}L(\vartheta_1) = 0, D^{(2,0)}L(\vartheta_2) = 0, D^{(0,2)}L(\vartheta_3) = 0,$

$D^{(0,2)}L(\vartheta_1) = 0, D^{(0,2)}L(\vartheta_2) = 0, D^{(0,2)}L(\vartheta_3) = 0,$

then the scheme (6) generates polynomial of degree 2. Further

$D^{(2,1)}L(\vartheta_1) = 0, D^{(2,1)}L(\vartheta_2) = 0, D^{(2,1)}L(\vartheta_3) = 0,$

$D^{(1,2)}L(\vartheta_1) = 0, D^{(1,2)}L(\vartheta_2) = 0, {}^{D(1,2)}L(\vartheta_3) = 0,$

$D^{(3,0)}L(\vartheta_1) = 0, D^{(3,0)}L(\vartheta_2) = 0, D^{(3,0)}L(\vartheta_3) = 0,$

$D^{(0,3)}L(\vartheta_1) = 0, D^{(0,3)}L(\vartheta_2) = 0, D^{(0,3)}L(\vartheta_3) = 0,$

so the scheme (6) generates polynomial of degree 3. Further more

$D^{(2,2)}L(\vartheta_1) = 0, D^{(2,2)}L(\vartheta_2) = 0, D^{(2,2)}L(\vartheta_3) = 0,$

$D^{(3,1)}L(\vartheta_1) = 0, D^{(3,1)}L(\vartheta_2) = 0, D^{(3,1)}L(\vartheta_3) = 0,$

$D^{(1,3)}L(\vartheta_1) = 0, D^{(1,3)}L(\vartheta_2) = 0, D^{(1,3)}L(\vartheta_3) = 0,$

$D^{(4,0)}L(\vartheta_1) = 0, D^{(4,0)}L(\vartheta_2) = 0, D^{(4,0)} L\vartheta_3) = 0$

$D^{(0,4)}L(\vartheta_1) = 0, D^{(0,4)}L(\vartheta_2) = 0, D^{(4,0)}L(\vartheta_3) = 0,$

so the scheme Equation (6) generates polynomial of degree 4.

Which completes the proof.

Theorem-2: For the parameter shift $(\tau_1, \tau_2) = \left(\frac{10}{4}, \frac{10}{4}\right)$ the subdivision scheme Equation (6) for $a = d = \frac{1}{16}$ and $b = c = \frac{3}{16}$, reproduces polynomial of degree 1 with respect to the parameterization defined in [26].

Proof: Let D^j with $j \in N^2$, denote a directional derivative. Since the symbol $L(z_1, z_2)$ satisfies the conditions in Theorem-1. Since $L(1,1) = 4$ and

$D^{(1,0)} L(1,1) - 4\tau_1 = 0, D^{(0,1)} L(1,1) - 4\tau_2 = 0$

then the scheme Equation (6) produced polynomial of degree 1.

Which completes the proof.

4. NUMERICAL EXAMPLES AND COMPARISON

In Fig. 1, we show the performance of bicubic 9-point bivariate subdivision scheme Equation (6). Fig. 1(a) is the initial control mesh. Fig. 1(b-c) showing the subdivision at first and second iteration. In Fig. 1(d), we get the limit surface after successive number of refinements.In Fig. 2, we show the performance of proposed 16-point bivariate scheme Equation (4). In Fig. 2(a), we design an initial control mesh. Fig. 2(b-c) showing the subdivision at first and second iteration. In Fig. 2(d), we get the limit surface after successive number of refinements.

5. CONCLUSION

This paper contributes towards the general bivariate parametric subdivision scheme for the surface modeling on the regular quad meshes. Some of the properties such that polynomial generation and polynomial reproduction of existing schemes are also calculated. We can observe that the approximating schemes have polynomial generation 6 and continuity 5 but the interpolating scheme have polynomial generation 3 and continuity 1. The general formula provides a variety of schemes to control the shape of initial mesh according to our own choice. By adjusting suitable value of parameters, we

TABLE 1. SHOWS THE CONTINUITY, POLYNOMIAL GENERATION AND POLYNOMIAL REPRODUCTION OF BIVARIATE SCHEMES

L	Schemes	Type	Continuity	Polynomial Generation	Polynomial Reproduction
1	9-Point Bivariate of [17]	Approximating	1	2	2
1	9-Point Bivariate of [18]		2	2	1
1	9-Point Bivariate of [5]		2	3	2
1	9-Point Bivariate of [2]		1	2	1
1	9-Point Bivariate of [7]	Interpolating	1	2	2
1	9-Point Bivariate of [19]	Approximating	2	2	1
1	9-Point Bivariate of [20]		2	2	1
2	16-Point Bivariate of [5]		4	5	1
2	9-Point Bivariate of [21]		1	2	2
3	16-Point Bivariate of [7]	Interpolating	1	3	3
3	16-Point Bivariate of [16]	Approximating	5	6	3
3	16-Point Bivariate of [18]		2	4	3
3	16-Point Bivariateof [22]		4	4	1
3	16-Point Bivariate of [16]		5	4	1
3	16-Point Bivariate of [23]		5	4	1
3	16-Point Bivariate of [6]		4	4	1
2	Proposed 16-Point Bivariate	Interpolating	1	2	1

(a) INITIAL MESH   *(b) FIRST ITERATION*   *(c) SECOND ITERATION*   *(d) LIMIT SURFACE*

FIG. 1. SHOWS THE INITIAL MESH, (B-D) SHOW THE DIFFERENT REFINEMENT STEPS

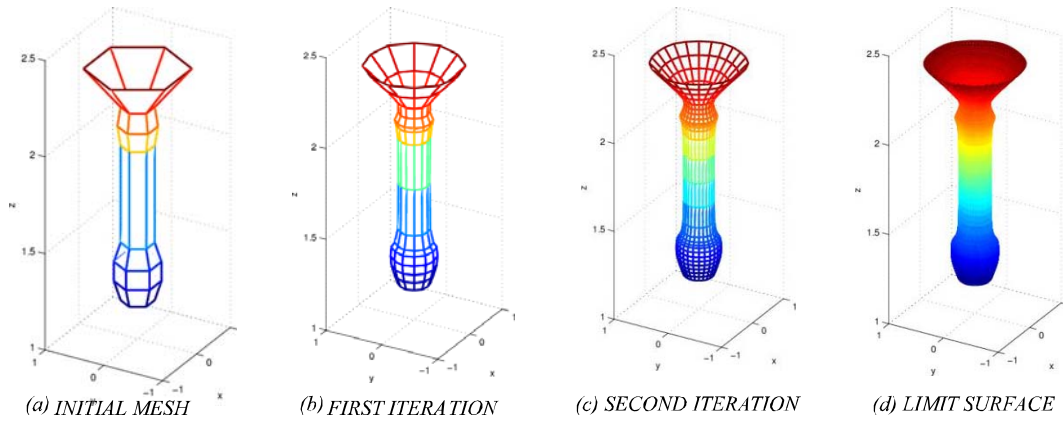*(a) INITIAL MESH*   *(b) FIRST ITERATION*   *(c) SECOND ITERATION*   *(d) LIMIT SURFACE*
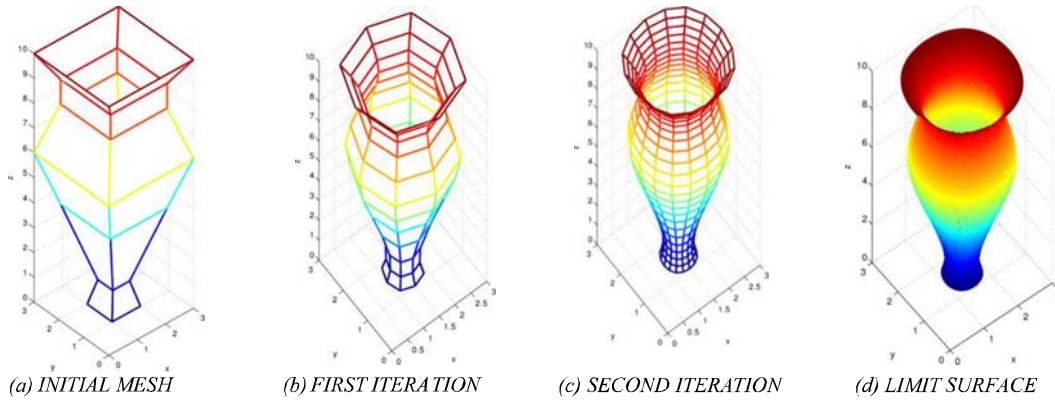
FIG. 2. SHOWS THE INITIAL MESH, (B-D) SHOW THE DIFFERENT REFINEMENT STEPS

can get suitable schemes to handle the initial mesh. Most of the existing tensor product schemes are the special case of proposed general bivariate schemes. Different snapshots show the geometrical appearance of initial meshes after the subdivision approach of bivariate schemes.

ACKNOWLEDGEMENT

REFERENCES

[1]   Rham, de.G., "Un Peude Mathematiques a Proposed' Une Courbe Plane", Revwed Mathematiques Elementry-II, Oevred Completes, pp. 678-689, 1947.

[2]   Chaikin, G.M., "An Algorithm for High-Speed Curve Generation", Computer Graphics and Image Processing, Volume 3, No. 4, pp. 346-349, 1974.

[3]   Dubuc, S., "Interpolation through an Iterative Scheme", Journal of Mathematical Analysis and Applications, Volume 114, pp. 185- 204, 1986.

[4]   Dyn, N., Levin, D., and Gregory, J.A., "A 4-Point Interpolatory Subdivision Scheme for Curve Design", Computer Aided Geometric Design, Volume 4, pp. 257-268, 1987.

Mehran University Research Journal of Engineering & Technology, Volume 37, No. 1, January, 2018 [p-ISSN: 0254-7821, e-ISSN: 2413-7219]

125

[5] Zheng, H.C., Huang, S.C., Guo, F., and Peng, G. H., "Designing Multi-Parameter Curve Subdivision Schemes with High Continuity", Applied Mathematics and Computation, Volume 243, pp. 197-208, 2014.

[6] Mustafa, G., Khan, F., and Ghaffar, A., "The M-Point Approximating Subdivision Scheme", Lobachevskii Journal of Mathematics, Volume 30, pp. 138-145, 2009.

[7] Shen, L., and Huang, Z., "A Class of Curve Subdivision Schemes with Several Parameters", Journal of Computer Aided Design and Computer Graphics, Volume 19, pp. 468-472, 2007.

[8] Mustafa, G., Ashraf, P., and Aslam, M., "Binary Univariate Dual and Primal Subdivision Schemes", SeMA Journal, Volume 65, pp. 22-35, 2014.

[9] Catmull, E., and Clark, J., "Recursively Generated B-Spline Surfaces on Arbitrary Topological Meshes", Computer Aided Geometric Design, Volume 10, pp. 350-355, 1978.

[10] Doo, D., and Sabin, M., "Behaviour of Recursive Division Surfaces Near Extraordinary Points", Computer Aided Design, Volume 10, No. 6, pp. 356-360, 1978.

[11] Lane, J.M., and Riesenfeld, R.F., "A Theatrical Development for Computer Generation and Display of Piecewise Polynomial Surfaces", IEEE Transaction on Pattern, Analysis and Machine Intelligence, Volume 2, No. 1, pp. 35-46, 1980.

[12] Dyn, N., Gregory, J., and Levin, D., "A Butterfly Subdivision Scheme for Surface Interpolation with Tension Control", ACM Transactions on Graphics, Volume 9, No. 2, pp. 160-169, 1990.

[13] Ghaffar, A., Mustafa, G., and Qin, K., "Construction and Application of 3-Point Tensor Product Scheme", Applied Mathematics, Volume 4, pp. 477-485, 2013.

[14] Kobbelt, L., "Interpolatory Subdivision on Open Quadrilateral Nets with Arbitrary Topology", Computer Graphics Forum, Proceedings of Eurographics, Volume 15, pp. 409-420, 1996.

[15] Zorin, D., Schroder, P., and Sweldens, W., "Interpolating Subdivision for Meshes with Arbitrary Topology", Computer Graphics Proceedings, ACM SIGGRATH, pp. 189–192, 1996.

[16] Mustafa, G., and Randhawa, A.H., "Complete Analysis of 3-Point Binary Subdivision Schemes", Journal of Pure and Applied Sciences, Volume 1-2, pp. 24-33, 2014.

[17] Mustafa, G., Ashraf, P., and Deng, J., "Generalized and Unified Families of Interpolating Subdivision Schemes", Numerical Mathematics Theory, Methods and Applications, Volume 7, pp. 193-213, 2014.

[18] Siddiqi, S.S., and Rehan, K., "Modified form of Binary and Ternary 3-Point Subdivision Schemes", Applied Mathematics and Computation, Volume 216, pp. 970-982, 2010.

[19] Ghaffar, A., Mustafa, G., and Qin, K., "The 4-Point á-Ary Approximating Subdivision Scheme", Open Journal of Applied Sciences, Volume 3, pp. 106-111, 2013.

[20] Siddiqi, S.S., and Ahmad, N., "A New Three-Point Approximating Subdivision Scheme", Applied Mathematics Letters, Volume 20, pp. 707-711, 2007.

[21] Hormanm, K., and Sabin, M., "A Family of Subdivision Schemes with Cubic Precision", Computer Aided Geometric Design, Volume 25, No. 1, pp. 41-52, 2008.

[22] Ghaffar, A., Mustafa, G., and Qin, K., "Unification and Application of 3-Point Approximating Subdivision Schemes of Varying Arity", Open Journal of Applied Sciences, Volume 2, pp. 48-52, 2012.

[23] Daniel, S., and Shunmugaraj, P., "Three Point Stationary and Non-Stationary Subdivision Schemes", 3rd International Conference on Geometric Modeling and Imaging, pp. 3-8, 2008.

[24] Siddiqi, S.S., and Younis, M., "Construction of M-Point Binary Approximating Subdivision Schemes", Applied Mathematics Letters, Volume 26, pp. 337-343, 2013.

[25] Ghaffar, A., "Lower and Higher Arity Subdivision Schemes and their Applications", PhD Thesis, Department of Mathematics, Islamia University of Bahawalpur, 2013.

[26] Romani, L., "A Chaikin-Based Variant of Lane-Riesenfeld Algorithm and its Non-Tensor Product Extension", Computer Aided Geometric Design, Volume 32, pp. 22-49, 2015.