Generalization of Binary Tensor Product Schemes Depends upon Four Parameters

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

RECEIVED ON 19.12.2016 ACCEPTED ON 21.02.2017

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
4-point binary subdivision scheme which generates \( C^1 \) 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 \((2n)\) and \((2n-1)\)-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. ALGORITHMS 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:

\[
\begin{align*}
    f_{i+1}^{k+1} &= \sum_{q=0}^{i+1} \alpha_{2q+1} f_{i+q}^k \\
    f_{2i+1}^{k+1} &= \sum_{q=0}^{2i} \alpha_{2q} f_{i+q}^k
\end{align*}
\]

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

\[ (1) \]
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

\[
\begin{align*}
\alpha_0 &= a C_{2l+1}^0 \\
\alpha_1 &= a C_{2l+1}^1 + b C_{2l+1}^0 \\
\alpha_2 &= a C_{2l+1}^2 + b C_{2l+1}^1 + c C_{2l+1}^0 \\
\alpha_3 &= a C_{2l+1}^3 + b C_{2l+1}^2 + c C_{2l+1}^1 + d C_{2l+1}^0 \\
& \vdots \\
\alpha_{2l} &= a C_{2l+1}^{2l} + b C_{2l+1}^{2l-1} + c C_{2l+1}^{2l-2} + d C_{2l+1}^{2l-3} \\
\alpha_{2l+1} &= a C_{2l+1}^{2l+1} + b C_{2l+1}^{2l} + c C_{2l+1}^{2l-1} + d C_{2l+1}^{2l-2} \\
\alpha_{2l+2} &= a C_{2l+1}^{2l+2} + b C_{2l+1}^{2l+1} + c C_{2l+1}^{2l} + d C_{2l+1}^{2l-1} \\
\alpha_{2l+3} &= a C_{2l+1}^{2l+3} + b C_{2l+1}^{2l+2} + c C_{2l+1}^{2l+1} + d C_{2l+1}^{2l} \\
\alpha_{2l+4} &= d
\end{align*}
\]

\(a, b, c, d\) are parameters, \(a + b + c + d = 1/\delta t\), \(l = 1, 2, 3, \ldots\) and

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

for \(r = 0, 1, 2, \ldots, 2l + 1\)

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:

\[
\begin{align*}
f_{21,2j}^{k+1} &= \sum_{q=0}^{l+2} \sum_{p=0}^{l+2} \alpha_{2q} \alpha_{2p+1} f_{i+q, j+p}^k \\
f_{21,2j+1}^{k+1} &= \sum_{q=0}^{l+2} \sum_{p=0}^{l+2} \alpha_{2q+1} \alpha_{2p} f_{i+q, j+p}^k \\
f_{21+1,2j}^{k+1} &= \sum_{q=0}^{l+2} \sum_{p=0}^{l+1} \alpha_{2q} \alpha_{2p+1} f_{i+q, j+p}^k \\
f_{21+1,2j+1}^{k+1} &= \sum_{q=0}^{l+2} \sum_{p=0}^{l+2} \alpha_{2q} \alpha_{2p} f_{i+q, j+p}^k \\
\end{align*}
\]

(3)

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 \(a = -\frac{1}{32}, b = \frac{1}{32}, c = \frac{1}{32}\) and \(d = 0\) 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:

\[
\begin{align*}
f_{2i,2j}^{k+1} &= \sum_{q=0}^{l+2} \sum_{p=0}^{l+2} \alpha_{2q} \alpha_{2p+1} f_{i+q, j+p}^k \\
f_{2i,2j+1}^{k+1} &= \sum_{q=0}^{l+2} \sum_{p=0}^{l+2} \alpha_{2q+1} \alpha_{2p} f_{i+q, j+p}^k \\
f_{2i+1,2j}^{k+1} &= \sum_{q=0}^{l+2} \sum_{p=0}^{l+1} \alpha_{2q} \alpha_{2p+1} f_{i+q, j+p}^k \\
f_{2i+1,2j+1}^{k+1} &= \sum_{q=0}^{l+2} \sum_{p=0}^{l+2} \alpha_{2q} \alpha_{2p} f_{i+q, j+p}^k \\
\end{align*}
\]

(5)
where the coefficients in Equation (5) can be calculated by replacing “\( l + \frac{1}{2} \)” in Equation (3), we get a system of equations with free parameters \( a, b, c, d \) such that
\[
a + b + c + d = \frac{1}{2^l}, \quad l = 2^l + 1
\]
where
\[
t \in \mathbb{Z}^+, \quad l = 0, 1, 2, \ldots, 2^t + 2.
\]
By substituting \( l = 0 \), we get general tensor product scheme as follows:

\[
l^{0}_{ij} = (a + b) f_i + (a + b) f_j + (a + b) f_{i + j} + (a + b)(a + b) f_{i + j} + (a + b)(a + b)(a + b) f_{i + j} + (a + b)(a + b)(a + b)(a + b) f_{i + j} + (a + b)(a + b)(a + b)(a + b)(a + b) f_{i + j} + (a + b)(a + b)(a + b)(a + b)(a + b)(a + b) f_{i + j} + \ldots
\]

\[
l^{1}_{ij} = (a + b + c + d) f_i + (a + b + c + d) f_j + (a + b + c + d) f_{i + j} + (a + b + c + d)(a + b + c + d) f_{i + j} + (a + b + c + d)(a + b + c + d)(a + b + c + d) f_{i + j} + (a + b + c + d)(a + b + c + d)(a + b + c + d)(a + b + c + d) f_{i + j} + \ldots
\]

\[
l^{2}_{ij} = (a + b + c + d + e) f_i + (a + b + c + d + e) f_j + (a + b + c + d + e) f_{i + j} + (a + b + c + d + e)(a + b + c + d + e) f_{i + j} + \ldots
\]

\[
(6)
\]

- By substituting \( a = b = 0 \) and \( c = d = \frac{1}{4} \) in Equation (6) we get 4-point tensor product approximating scheme of [2].
- By substituting \( a = c = \frac{1}{8} \) and \( b = d = \frac{1}{4} \) in Equation (6) we get 9-point tensor product approximating scheme of [5].
- By substituting \( a = \frac{1}{16} \) and \( b = c = \frac{3}{16} \) in Equation (6) we get 9-point tensor product interpolating scheme of [7].
- By substituting \( a = d = \frac{3}{32} \) and \( b = \frac{11}{32} \) in Equation (6) we get 9-point tensor product approximating scheme of [20].
- By substituting \( a = d = -\frac{1}{32} \) and \( b = c = \frac{7}{32} \) in Equation (6) we get 9-point tensor product approximating scheme of [21].
- By taking \( a = d = \frac{\mu}{16} \) and \( b = c = \frac{4 - \mu}{16} \) in Equation (6) we get 9-point tensor product approximating scheme of [22].
- By taking \( a = d = -3\omega \) and \( b = c = \frac{1}{4} + 3\omega \) 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 \( a = \frac{1}{16}, \quad b = \frac{1}{4} \) and \( d = 0 \) in Equation (5) we get 16-point tensor product interpolating scheme of [4].
- By taking \( a = d = \frac{1 + 6\omega}{96} \) and \( b = c = \frac{5 + 6\omega}{96} \) in Equation (5) we get 16-point tensor product approximating scheme of [6].
- Substituting \( a = d = \frac{1}{64} \) and \( b = c = \frac{3}{64} \) in Equation (5) we get 16-point tensor product approximating scheme of [19].
- For \( a = d = \frac{\omega}{16}, \quad b = c = \frac{1 - \omega}{16} \) in Equation (5) we get 16-point tensor product approximating scheme of [19].
- By putting \( a = d = \frac{5}{128} \) and \( b = c = \frac{13}{128} \) 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 \( a = d = \frac{u}{64} \) and \( b = c = \frac{4 + u}{64} \) 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 \( j \in \mathbb{N}^2 \), denote a directional derivative. Since \( L(1, 1) = 4 \) and

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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 \( j \in \mathbb{N}^2 \), 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>
<thead>
<tr>
<th>L</th>
<th>Schemes</th>
<th>Type</th>
<th>Continuity</th>
<th>Polynomial Generation</th>
<th>Polynomial Reproduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9-Point Bivariate of [17]</td>
<td>Approximating</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>9-Point Bivariate of [18]</td>
<td></td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>9-Point Bivariate of [5]</td>
<td></td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>9-Point Bivariate of [2]</td>
<td></td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>9-Point Bivariate of [7]</td>
<td>Interpolating</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>9-Point Bivariate of [19]</td>
<td></td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>9-Point Bivariate of [20]</td>
<td></td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>16-Point Bivariate of [5]</td>
<td></td>
<td>4</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>9-Point Bivariate of [21]</td>
<td></td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>16-Point Bivariate of [7]</td>
<td>Interpolating</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>16-Point Bivariate of [16]</td>
<td></td>
<td>5</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>16-Point Bivariate of [18]</td>
<td></td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>16-Point Bivariate of [22]</td>
<td>Approximating</td>
<td>4</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>16-Point Bivariate of [16]</td>
<td></td>
<td>5</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>16-Point Bivariate of [23]</td>
<td></td>
<td>5</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>16-Point Bivariate of [6]</td>
<td></td>
<td>4</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Proposed 16-Point Bivariate</td>
<td>Interpolating</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
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

This work is supported by The Islamia University, Bahawalpur, Pakistan, NRPU Project No. 3183 the Indigenous Ph.D. 5000 Fellowship Program of Higher Education Commission, Islamabad.

REFERENCES


Generalization of Binary Tensor Product Schemes Depends upon Four Parameters


