

Construction and Application of Subdivision Surface Scheme Using Lagrange Interpolation Polynomial

Faheem Khan*, Noreen Batool, Iram Mukhtar

Department of Mathematics, University of Sargodha, Sargodha, Pakistan

Email: *fahimscholar@gmail.com, noreen_batool@yahoo.com, irammukhtar1133@hotmail.com

Received April 22, 2013; revised May 22, 2013; accepted May 30, 2013

Copyright © 2014 Faheem Khan *et al.* This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. In accordance of the Creative Commons Attribution License all Copyrights © 2014 are reserved for SCIRP and the owner of the intellectual property Faheem Khan *et al.* All Copyright © 2014 are guarded by law and by SCIRP as a guardian.

ABSTRACT

This paper offers a general formula for surface subdivision rules for quad meshes by using 2-D Lagrange interpolating polynomial [1]. We also see that the result obtained is equivalent to the tensor product of $(2N + 4)$ -point n -ary interpolating curve scheme for $N \geq 0$ and $n \geq 2$. The simple interpolatory subdivision scheme for quadrilateral nets with arbitrary topology is presented by L. Kobbelt [2], which can be directly calculated from the proposed formula. Furthermore, some characteristics and applications of the proposed work are also discussed.

KEYWORDS

Subdivision Scheme; Interpolating Subdivision Scheme; Tensor Product Scheme; Auxiliary Points; Lagrange Interpolation Polynomial

1. Introduction

There are two general classes of subdivision schemes, namely, approximating and interpolating schemes. The limit curve of an approximating scheme usually does not pass through the control points of control polygon. As the level of refinement increases, the polygon usually shrinks towards the final limit curve. The interpolating schemes are more attractive than approximating schemes because of their interpolation property. All vertices in the control polygon are located on the limit curve of the interpolation scheme, which facilitates and simplifies the graphics algorithms and engineering designs.

Lian generalized the classical binary 4-point and 6-point interpolatory subdivision schemes to a -ary setting for any integer $a \geq 3$. After that, the a -ary 3-point and 5-point interpolatory subdivision schemes for curve design for arbitrary odd integer $a \geq 3$ [3,4] were introduced. After that, Lian [5] investigated both the $2m$ -point, a -ary for any $a \geq 2$ and $(2m + 1)$ -point, a -ary for any odd $a \geq 3$ interpolatory subdivision schemes for curve design. Ko [6] presented explicitly a new formula for the mask of $(2N + 4)$ -point binary interpolating and approximating subdivision schemes with two parameters. The proposed work presents a new observation about the curve case given by Najma [7]. In this work, we avoid finding the mask of subdivision schemes separately, as a result, its approach is simple and avoids complex computation when deriving subdivision rules.

The rest of the paper is organized as follows. Section 2 gives some preliminaries results and a new relation for $(2N + 4)$ -point n -ary interpolating curve scheme for closed and open polygon to access main result. Section 3 presents the construction for general formula of the surface case using Lagrange interpolating polynomial, and some characteristics are also discussed. In Section 4, we also give some numerical examples for the visual performance of the proposed work. This work also provides some special cases of the classical subdivision schemes.

*Corresponding author.

2. Preliminary Results

Let \mathbb{Z} be the set of integers and $a = \{a_i | i \in \mathbb{Z}\}$ a set of constants. The general form of univariaten-ary subdivision scheme which maps a polygon $f^k = \{f_i^k\}_{i \in \mathbb{Z}}$ is defined by

$$f_{ni+s}^{k+1} = \sum_{j \in \mathbb{Z}} a_{nj+s} f_{i-j}^k, \quad s = 0, 1, 2, \dots, n-1, \tag{2.1}$$

where the set $a = \{a_i | i \in \mathbb{Z}\}$ of coefficients is called mask of the subdivision scheme. A necessary condition for the uniform convergence of the subdivision scheme is

$$\sum_{j \in \mathbb{Z}} a_{nj+s} = 1, \quad s = 0, 1, 2, \dots, n-1, \tag{2.2}$$

Let Ω_{2N+1} be the space of all polynomials of degree $\leq 2N+1$. Where, N is a non-negative integer. If $\{L_\mu(x)\}_{\mu=-N}^{N+1}$ is fundamental Lagrange polynomial corresponding to the nodes $\{\mu\}_{\mu=-N}^{N+1}$ defined by

$$L_\mu(x) = \prod_{j=-N, j \neq \mu}^{N+1} \frac{x-j}{\mu-j} \tag{2.3}$$

for which

$$L_\mu(j) = \delta_{\mu-j}, \quad \mu, j = -N, \dots, N+1$$

and

$$\sum_{\mu=-N}^{N+1} p(\mu) L_\mu(x) = p(x), \quad p \in \Omega_{2N+1}$$

where, $\delta_{\mu,j}$ is the Kronecker delta, defined as

$$\delta_{\mu,j} = \begin{cases} 1, & \mu = j \\ 0, & \mu \neq j \end{cases} \tag{2.4}$$

Using all the above mentioned identities Ko [6] presented the general formula for the mask of $(2N+4)$ -point binary interpolating symmetric subdivision schemes. After that Najma [7] generalized the result for $(2N+4)$ -point n -ary interpolating symmetric subdivision scheme and gave the following formula for the mask of n -ary interpolating schemes.

$$\begin{cases} a_{n,j} = \delta_{j,0} - v \xi_2(N, j), \\ a_{nj+s} = \xi_1(N, j, n, s) - a_{n(N+1)+s} \xi_2(N, j) - a_{n(N+1)+t} \xi_3(N, j), \end{cases} \tag{2.5}$$

where

$$\begin{aligned} \xi_1(N, j, n, s) &= \frac{\prod_{b=-N-1}^N (ni+s)}{n^{2N+1} (-1)^{-j+N-1} (s+nj)(N-j)!(N+1+j)!}, \\ \xi_2(N, j) &= \frac{(-1)^{(j+N)} (2N+2)!}{(N-j)!(N+j+1)!(N-j+1)!}, \\ \xi_3(N, j) &= \frac{(-1)^{j+N+1} (2N+2)!}{(N-j)!(N+j+1)!(N-j+2)!}, \end{aligned}$$

and

$$a_{n(N+1)+s} = a_{-n(N+2)-s} \tag{2.6}$$

The free parameters $a_{n(N+1)+s}$ can be explicitly defined as

$$a_{n(N+1)+s} = \frac{(d-n)(d-2n)\cdots(d-(2Nn+3n))}{(-1)^{(2N+3)}(n)^{(2N+3)}(2N+3)!},$$

where $d = n(N+1) + s$.

Here, n stands for n -ary subdivision scheme (i.e. $n = 2$ (binary), 3 (Ternary), 4 (quaternary)⋯), $N \geq 0$, $j = -N-1, \dots, N$, $s = 0, 1, 2, \dots, n-1$ and $t = n-s$. Considering the symmetry of the scheme and construction of the mask formula described above, $(2N+4)$ -point interpolating subdivision schemes are presented in the following form

$$f_{2i+\alpha}^{k+1} = \sum_{l=-N-1}^{N+2} a_{2l-\alpha} f_{i+l}^k, \text{ where } \alpha = 0, 1. \tag{2.7}$$

Here, $N \geq 0$ with the symmetry condition is

$$a_{-2(N+1)-\alpha} = a_{2(N+1)+\alpha}$$

Setting $a_{2(N+2)} = 0$ and $a_{2(N+2)} = v$, the mask $a_{2l-\alpha}$ of the schemes comes from the generalized formula for the mask of $(2N+4)$ -point interpolating schemes (2.5). Following the procedure of binary case, we have derived the following form of $(2N+4)$ -point ternary interpolating subdivision schemes are presented in the following form

$$f_{3i+\alpha}^{k+1} = \sum_{l=-N-1}^{N+2} a_{3l-\alpha} f_{i+l}^k, \text{ where } \alpha = 0, 1, 2. \tag{2.8}$$

Here, $N \geq 0$ and for the symmetry condition, $a_{-3(N+1)-\alpha} = a_{3(N+1)+\alpha}$

Setting $a_{3(N+2)} = 0$ and $a_{3(N+2)} = v$, the mask $a_{3l-\alpha}$ is calculated from the same mask formula (2.5). In the same way, $(2N+4)$ -point quaternary interpolating subdivision scheme has the form

$$f_{4i+\alpha}^{k+1} = \sum_{l=-N-1}^{N+2} a_{4l-\alpha} f_{i+l}^k, \text{ where } \alpha = 0, 1, 2, 3. \tag{2.9}$$

where, $N \geq 0$ and for the symmetry of the scheme, $a_{-4(N+1)-\alpha} = a_{4(N+1)+\alpha}$.

Setting $a_{4(N+2)} = 0$ and $a_{4(N+2)} = v$. Finally, from (2.7)-(2.9), $(2N+4)$ -point n -ary interpolating schemes has the following form

$$f_{ni+\alpha}^{k+1} = \sum_{l=-N-1}^{N+2} a_{nl-\alpha} f_{i+l}^k, \text{ where } \alpha = 0, 1, 2, \dots, n-1, \tag{2.10}$$

With $N \geq 0, n \geq 2$, and for the following symmetry condition

$$a_{-n(N+1)-\alpha} = a_{n(N+1)+\alpha} \tag{2.11}$$

where $a_{n(N+2)} = 0$ and $a_{n(N+2)} = v$.

Construction of the Schemes for Open Polygon

When dealing with open initial polygon $f^0 = \{f_i^0 : i = 0, \dots, N\}$, it is not possible to refine the first and last edges by rules (2.10) for interpolating subdivision schemes. However the extension of this strategy to deal with open polygon requires a well-define neighborhood of end points. Since the first and last edges can be treated analogously, it will be sufficient to derive the rules only for one side of the polygon. To this aim define the auxiliary point $f_{-i}^0 = 2f_0^0 - f_i^0$ as extrapolatory rule in the initial polygon f^0 . Then the nonrefined open polygon $\{f_i^k : i = 0, \dots, n^k N\}$ can be refined by the rules defined below. The formula described in (2.10) for interpolating scheme is not helpful to refine first and last edges of open polygon. Then to refine the open polygon by

$(2N + 4)$ -point interpolating scheme using auxiliary points is defined as following

$$f_{ni+\alpha}^{k+1} = \sum_{l=-N-1}^{-i-1} \left(2a_{nl-\alpha} f_0^k - a_{nl-\alpha} f_{-(i+1)}^k \right) + \sum_{l=-i}^{b+2} a_{nl-\alpha} f_{i+1}^k, \tag{2.12}$$

where $\alpha = 0, 1, 2, \dots, (n-1)$, $N \geq 0, i = 0, 1, \dots, N$ and $n \geq 2$, where, the weights satisfies the same condition (2.11).

Example: If an open polygon is refined by using the 6-point ternary interpolating subdivision scheme using (2.10), then two auxiliary points $f_{-2}^0 = f_0^0 - f_2^0$ and $f_{-1}^0 = 2f_0^0 - f_1^0$ has to be defined in the coarsest polygon f^0 . The first two edges $f_0^k f_1^k$ and $f_1^k f_2^k$ of the nonrefined polygon $\{f_i^k : i = 0, \dots, 3^k N\}$ can be refined by the rules that can be calculated directly by (2.12). Substituting $n = 3, N = 1$ in (2.12),

$$f_{3i+\alpha}^{k+1} = \sum_{l=-2}^{-i-1} \left(2a_{3l-\alpha} f_0^k - a_{3l-\alpha} f_{-(i+1)}^k \right) + \sum_{l=0}^3 a_{3l-\alpha} f_{i+1}^k,$$

where $\alpha = 0, 1, 2, i = 0, 1$.

Then, for $i = 0$,

$$\begin{aligned} f_0^{k+1} &= (2a_{-6} + 2a_{-3} + a_0) f_0^k + (a_3 - a_{-3}) f_1^k + (a_6 - a_{-6}) f_2^k + a_9 f_3^k, \\ f_1^{k+1} &= (2a_{-7} + 2a_{-4} + a_{-1}) f_0^k + (a_2 - a_{-4}) f_1^k + (a_5 - a_{-7}) f_2^k + a_8 f_3^k, \\ f_2^{k+1} &= (2a_{-8} + 2a_{-5} + a_{-2}) f_0^k + (a_1 - a_{-5}) f_1^k + (a_4 - a_{-8}) f_2^k + a_7 f_3^k. \end{aligned}$$

For, $i = 1$

$$\begin{aligned} f_3^{k+1} &= (2a_{-6} + a_{-3}) f_0^k + (a_0 - a_{-6}) f_1^k + a_3 f_2^k + a_6 f_3^k + a_9 f_4^k, \\ f_4^{k+1} &= (2a_{-7} + a_{-4}) f_0^k + (a_{-1} - a_{-7}) f_1^k + a_2 f_2^k + a_5 f_3^k + a_8 f_4^k, \\ f_5^{k+1} &= (2a_{-8} + a_{-5}) f_0^k + (a_{-2} - a_{-8}) f_1^k + a_1 f_2^k + a_4 f_3^k + a_7 f_4^k. \end{aligned}$$

3. Tensor Product of $(2N + 4)$ -Point Interpolating Subdivision Scheme

Given a set of control points $p_{i,j}^k \in R^N, i, j \in Z, N \geq 2$, where k is a non-negative integer indicates the subdivision level. n -ary subdivision surface is tensor product of n -ary subdivision curve defined by

$$p_{ni+\alpha, nj+\beta}^{k+1} = \sum_{r=0}^m \sum_{s=0}^m a_{\alpha,r} a_{\beta,s} p_{i+r, j+s}^k, \quad \alpha, \beta = 0, 1, \dots, n-1, \tag{3.1}$$

where, $a_{\alpha,r}$ satisfies

$$\sum_{j=0}^m a_{\alpha,j} = 1, \quad \alpha = 0, 1, \dots, n-1. \tag{3.2}$$

Given initial values $p_{i,j}^0 \in R^N, i, j \in Z, i, j \in \mathbb{Z}$, then in the limit $k \rightarrow \infty$, the process (3.1) defines an infinite set of points in R^N . The sequence of values $\{p_{i,j}^k\}$ is related, in a natural way, with a diadic mesh points $\left(\frac{i}{n^k}, \frac{j}{n^k}\right), i, j \in Z$. The process then defines a scheme whereby $p_{ni+\alpha, nj+\beta}^{k+1}$ replaces the value $p_{i+\alpha/n, j+\beta/n}$ at the mesh point $\left(\frac{i+\alpha/n}{n^k}, \frac{j+\beta/n}{n^k}\right)$ for $\alpha, \beta \in \{0, n\}$, while the values $p_{ni+\alpha, nj+\beta}^{k+1}$ are inserted at the new mesh points $\left(\frac{ni+\alpha}{n^{k+1}}, \frac{nj+\beta}{n^{k+1}}\right)$ for $\alpha, \beta = 0, 1, \dots, n-1$ (where α and β are not zero at the same time). Labeling of old and new points is shown in **Figure 1**, which illustrates subdivision schemes (3.1).

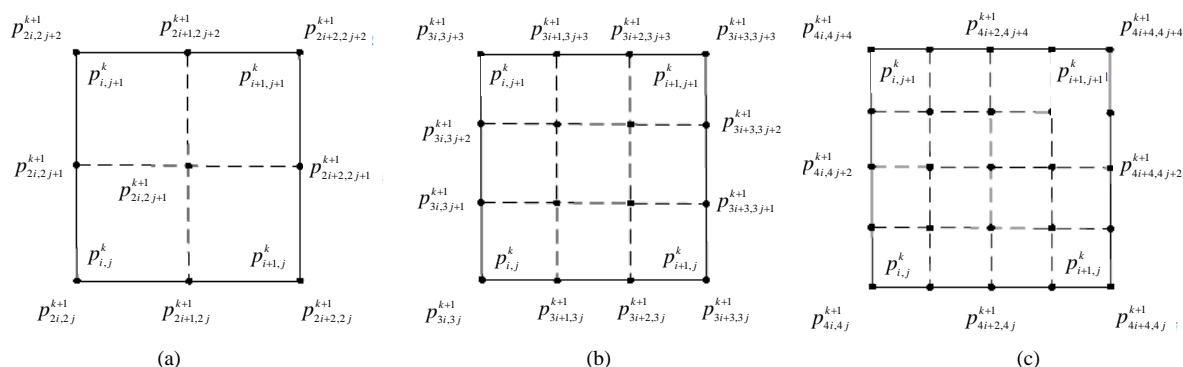


Figure 1. Solid lines show one face of coarse polygons whereas dotted lines are refined polygons. (a)-(c) can be obtained by subdividing one face into four, nine and sixteen new faces by using (3.1) for $n = 2,3,4$ respectively.

Construction

Let \mathbb{Z} be the set of integers and the space of all polynomials of degrees $\leq 2\gamma + 1$ and $\leq 2\sigma + 1$ is denoted by $\Omega_{2\gamma+1}$ and $\Omega_{2\sigma+1}$ respectively. If $\{L_\mu(x)\}_{\mu=-\gamma}^{\gamma+1}$ are fundamental Lagrange interpolating polynomials corresponding to the nodes $\{\mu\}_{\mu=-\gamma}^{\gamma+1}$ and $\{\nu\}_{\nu=-\sigma}^{\sigma+1}$. The Lagrange interpolation polynomial for tensor product case is defined as [1],

$$p(x, y) = \sum_{\mu=-\gamma}^{\gamma+1} \sum_{\nu=-\sigma}^{\sigma+1} L_{\mu,\nu}(x, y) p(\mu, \nu),$$

where

$$L_{\mu,\nu}(x, y) = \prod_{i=-\gamma, i \neq \mu}^{\gamma+1} \frac{x-i}{\mu-i} \times \frac{x-j}{\nu-j}, \tag{3.3}$$

$$\mu = -\gamma, \dots, \gamma+1, \quad \nu = -\sigma, \dots, \sigma+1.$$

$\delta_{\mu,i}$ and $\delta_{\nu,j}$ are Kroneker delta symbols defined as,

$$\delta_{\mu,i} = \begin{cases} 1, & \mu = i \\ 0, & \mu \neq i \end{cases}$$

and

$$\delta_{\nu,i} = \begin{cases} 1, & \nu = i \\ 0, & \nu \neq i \end{cases}$$

Here, some important results for the formulation of required form of tensor product scheme can be verified using (2.3). That is for each $\mu = -\gamma-1, \dots, \gamma$ and $\nu = -\sigma-1, \dots, \sigma$ (Using the result [1]),

$$L_{-i}(-\gamma-1) = \frac{(-1)^{i+\gamma} (2\gamma+2)!}{(\gamma-i)!(\gamma+i+1)!(\gamma-i+1)},$$

$$L_{-j}(-\sigma-1) = \frac{(-1)^{j+\sigma} (2\sigma+2)!}{(\sigma-j)!(\sigma+j+1)!(\sigma-j+1)},$$

$$L_{-i}(\gamma+2) = \frac{(-1)^{i+\gamma+1} (2\gamma+2)!}{(\gamma-i)!(\gamma+i+1)!(\gamma+i+2)},$$

$$L_{-j}(\sigma + 2) = \frac{(-1)^{j+\sigma+1} (2\sigma + 2)!}{(\sigma - j)!(\sigma + j + 1)!(\sigma + j + 2)},$$

$$L_{-i}\left(\frac{2s_1 + 1}{2n}\right) = \frac{\prod_{b=-(\gamma+1)}^{\gamma} (2nb + 2s_1 + 1)}{(2n)^{2\gamma+1} (-1)^{-i+\gamma-1} (2ni + 2s_1 + 1)(\gamma - i)!(\gamma + i + 1)!},$$

$$L_{-j}\left(\frac{2s_2 + 1}{2m}\right) = \frac{\prod_{b=-(\sigma+1)}^{\sigma} (2mb + 2s_2 + 1)}{(2m)^{2\sigma+1} (-1)^{-j+\sigma-1} (2mj + 2s_2 + 1)(\sigma - j)!(\sigma + j + 1)!},$$

$$L_{-i}\left(\frac{s_1}{n}\right) = \frac{\prod_{b=-\gamma-1}^{\gamma} (nb + s_1)}{n^{2\gamma+1} (-1)^{-i+\gamma-1} (ni + s_1)(\gamma - i)!(\gamma + 1 + i)!},$$

$$L_{-j}\left(\frac{s_2}{m}\right) = \frac{\prod_{b=-\sigma-1}^{\sigma} (mb + s_2)}{m^{2\sigma+1} (-1)^{-j+\sigma-1} (mj + s_2)(\sigma - j)!(\sigma + 1 + j)!}.$$

The mask of a subdivision scheme shows the contribution of a single original vertex to each new, subdivided vertex. To find the mask of a scheme, we need to find all ways to get from the origin to each point in the grid. For the tensor product scheme, this is simply the tensor product of the univariate case.

Lemma 3.1. [8] *Given initial control polygon $p_{i,j}^0 = p_{i,j}$, $i, j \in Z$, let the values $p_{i,j}^k$, $k \geq 1$ be defined recursively by subdivision process (3.1) together with (3.2) then the scheme derived by tensor product naturally get four-sided support region.*

It can be loosely say that the support is the tensor product of the supports of the two regions, just as one can loosely say that Doo-Sabin is the generalization of the tensor product of two Chaikin constructions.

Lemma 3.2. [9] *Given initial control polygon $p_{i,j}^0 = p_{i,j}$, $i, j \in Z$, let the values $p_{i,j}^k$, $k \geq 1$ be defined recursively by subdivision process (3.1) together with (3.2), then if a scheme is derived from a tensor product, then the level of continuity can be determined between pieces by reference to the underlying basis functions, i.e. all the tensor product schemes have the same continuity as their counterparts.*

The general formula which generates the mask $\{a_i\}_{i=-2\gamma-3}^{2\gamma+3}$ and $\{a_j\}_{j=-2\sigma-3}^{2\sigma+3}$ of n-ary approximating schemes presented by [7] is

$$\begin{cases} a_{ni} = \delta_{i,0} - \nu \xi_2(\gamma, j) \\ a_{ni+s_1} = \xi_1(\gamma, i, n, s_1) - a_{n(\gamma+1)+s_1} \xi_2(\gamma, i) - a_{n(\gamma+1)+t_1} \xi_3(\gamma, i), \end{cases} \tag{3.4}$$

and

$$\begin{cases} a_{mj} = \delta_{j,0} - \nu \eta_2(\sigma, j) \\ a_{mj+s_2} = \eta_1(\sigma, j, n, s_2) - a_{n(\sigma+1)+s_2} \eta_2(\eta, j) - a_{n(\sigma+1)+t_2} \eta_3(\sigma, j), \end{cases} \tag{3.5}$$

where

$$\xi_1(\gamma, i, n, s_1) = \frac{\prod_{b=-(\gamma+1)}^{\gamma} (nb + s_1)}{(n)^{2\gamma+1} (-1)^{-i+\gamma-1} (ni + s_1)(\gamma - i)!(\gamma + i + 1)!},$$

$$\eta_1(\sigma, j, m, s_2) = \frac{\prod_{b=-(\sigma+1)}^{\sigma} (mb + s_2)}{(m)^{2\sigma+1} (-1)^{-j+\sigma-1} (mj + s_2)(\sigma - j)!(\sigma + j + 1)!},$$

$$\xi_2(\gamma, i) = \frac{(-1)^{i+\gamma} (2\gamma + 2)!}{(\gamma - i)!(\gamma + i + 1)!(\gamma - i + 1)!},$$

$$\eta_2(\sigma, j) = \frac{(-1)^{j+\sigma} (2\sigma + 2)!}{(\sigma - j)!(\sigma + j + 1)!(\sigma - j + 1)},$$

$$\xi_3(\gamma, i) = \frac{(-1)^{i+\gamma+1} (2\gamma + 2)!}{(\gamma - i)!(\gamma + i + 1)!(\gamma + i + 2)}$$

$$\eta_3(\sigma, j) = \frac{(-1)^{j+\sigma+1} (2\sigma + 2)!}{(\sigma - j)!(\sigma + j + 1)!(\sigma + j + 2)}.$$

Here, n, m stands for n -ary, m -ary subdivision schemes respectively (i.e $n, m = 2$ (binary), 3(ternary), 4(quaternary) ...), $\gamma, \sigma \geq 0$ $i = -\gamma - 1, \dots, \gamma, j = -\sigma - 1, \dots, \sigma, s_1 = 1, 2, \dots, n - 1, s_2 = 1, 2, \dots, m - 1, t_1 = n - s_1$, and $t_2 = m - s_2$

The free parameter $a_{n(\gamma+1)+s_1}$ and $a_{m(\sigma+1)+s_2}$ are defined as

$$a_{n(\gamma+1)+s_1} = \frac{(d - n)(d - 2n) \dots (d - (2\gamma n + 3n))}{(-1)^{2\gamma+3} (n)^{2\gamma+3} (2\gamma + 3)!}, \tag{3.6}$$

where $d = n(\gamma + 1) + s_1$ and

$$a_{m(\sigma+1)+s_2} = \frac{(e - m)(e - 2m) \dots (e - (2\sigma m + 3m))}{(-1)^{2\sigma+3} (2m)^{2\sigma+3} (2\sigma + 3)!}, \tag{3.7}$$

where $e = m(\sigma + 1) + s_2$.

As each mask $a_{i,j}$ of the refinement rule satisfies $a_{i,j} = b_i b_j$, where b_i, b_j are the mask of univariate subdivision schemes, then

$$a_{(ni+s_1, mj+s_2)} = a_{ni+s_1} a_{mj+s_2}.$$

The tensor product of $(2N + 4)$ -point interpolating subdivision scheme is presented as,

$$f_{ni+\alpha, mj+\beta}^{k+1} = \sum_{l_1=-\gamma-1}^{\gamma+2} \sum_{l_2=-\sigma-1}^{\sigma+2} a_{(nl_1-\sigma, ml_2-\beta)} f_{i+l_1, j+l_2}^k, \tag{3.8}$$

where, $\alpha = 0, 1, 2, \dots, (n - 1), \beta = 0, 1, 2, \dots, (m - 1), \gamma, \sigma \geq 0$, and $n, m \geq 2$ and symmetry conditions are,

$$\begin{cases} a_{-n(\gamma+1)-\alpha} = a_{n(\gamma+1)+\alpha} \\ a_{-m(\sigma+1)-\beta} = a_{m(\sigma+1)+\beta}. \end{cases} \tag{3.9}$$

Taking $a_{n(\gamma+2)} = a_{m(\sigma+2)} = 0$, and the constants $a_{nl_1-1-\alpha}$ and $a_{ml_2-1-\beta}$ could be evaluated by using the results (3.4) and (3.5).

Example: Consider the tensor product of the 4-point DD interpolating subdivision scheme, while DD scheme can be calculated using the result (2.5) mentioned in Section 2. The Laurent polynomial of the scheme is given as

$$a(z) = \frac{1}{16}(-z^{-3} + 9z^{-1} + 1 + 9z^1 - 1z^3).$$

This implies

$$a(z_1) = \frac{1}{16}(-z_1^{-3} + 9z_1^{-1} + 1 + 9z_1 - z_1^3),$$

$$a(z_2) = \frac{1}{16}(-z_2^{-3} + 9z_2^{-1} + 1 + 9z_2 - z_2^3).$$

Since, $a(z_1, z_2) = a(z_1)a(z_2)$ then, we can obtain the Laurent polynomial of the 4-point tensor product binary interpolating scheme S_a . So that the suggested 4-point tensor product binary interpolating scheme is

$$\left\{ \begin{aligned} f_{2i,2j}^{k+1} &= f_{i,j}^k, \\ f_{2i+1,2j}^{k+1} &= \frac{1}{16}(-f_{i-1,j}^k + 9f_{i,j}^k + 9f_{i+1,j}^k - f_{i+2,j}^k), \\ f_{2i,2j+1}^{k+1} &= \frac{1}{16}(-f_{i,j-1}^k + 9f_{i,j}^k + 9f_{i,j+1}^k - f_{i,j+2}^k), \\ f_{2i+1,2j+1}^{k+1} &= \frac{1}{256}(f_{i-1,j-1}^k - 9f_{i,j-1}^k - 9f_{i+1,j-1}^k + f_{i+2,j-1}^k \\ &\quad - 9f_{i-1,j}^k + 81f_{i,j}^k + 81f_{i+1,j}^k - 9f_{i+2,j}^k - 9f_{i-1,j+1}^k \\ &\quad + 81f_{i,j+1}^k + 81f_{i+1,j+1}^k - 9f_{i+2,j+1}^k + f_{i-1,j+2}^k \\ &\quad - 9f_{i,j+2}^k - 9f_{i+1,j+2}^k + f_{i+2,j+2}^k) \end{aligned} \right. \tag{3.10}$$

Using the result obtained above for the tensor product of interpolatory scheme (3.8), tensor product of 4-point DD scheme can be calculated directly. Since the DD scheme has C^1 continuity, then by lemma (3.2) its tensor product has the same continuity. Substituting $n = m = 2$ in (3.8) and (3.9), $\alpha, \beta = 0, 1$ and $\gamma = \sigma = 0$, the symmetry conditions becomes

$$\begin{cases} a_{-2-\alpha} = a_{2+\alpha}, \\ a_{-2-\beta} = a_{2+\beta}, \end{cases} \tag{3.11}$$

then formula (3.8) attains the form,

$$f_{2i+\alpha,2j+\beta}^{k+1} = \sum_{l_1=-1}^2 \sum_{l_2=-1}^2 a_{(2l_1-\sigma,2l_2-\beta)} f_{i+l_1,j+l_2}^k, \tag{3.12}$$

As each mask $a_{i,j}$ of the refinement rule satisfies $a_{i,j} = b_1 b_j$, then

$$a_{(2l_1-\sigma,2l_2-\beta)} = a_{2l_1-\alpha} a_{2l_2-\beta}$$

As $n = m = 2$ so $a_4 = 0$ for both n and m , when $v_1 = v_2 = 0$, $\omega_1 = \omega_2 = \frac{-1}{16}$, $b_1 = b_2 = 0$ and $s_1, s_2 = 1$ is substituted in (3.8) and (3.9) our requirement is fulfilled, that is the rules (3.10) are obtained.

Example: A simple interpolatory subdivision scheme for quadrilateral nets with arbitrary topology is presented by L. Kobbelt [2] which generates C^1 surfaces in the limit. In the first step they present the refinement rules derived by the modification of the well-known Dyn *et al.* [10] 4-point interpolatory subdivision scheme for curve design. The natural way to define refinement operators for quadrilateral nets is therefore to modify a tensor product scheme such that special rules for the vicinity of non regular vertices are found.

The modified form of Dyn scheme can be evaluated by setting the value of $n = 2, N = v = 0, a_4 = 0$ and $a_3 = -\omega/16$ (where $0 < \omega < 2(\sqrt{5} - 1)$) in (2.10) and (2.11), the following refinement rules are obtained

$$\left\{ \begin{aligned} f_{2i}^{k+1} &= f_i^k, \\ f_{2i+1}^{k+1} &= \frac{8+\omega}{16}(f_i^k + f_{i+1}^k) - \frac{\omega}{16}(f_{i-1}^k + f_{i+2}^k). \end{aligned} \right. \tag{3.13}$$

They used the simple tensor product as the basis for the modification of refinement rules of irregular quadrilateral nets. Since it is interpolating scheme, so $f_{2i,2j}^{k+1} = f_{i,j}^k$ and the edge points $f_{2i+1,2j}^{k+1}$ and $f_{2i,2j+1}^{k+1}$ are given as,

$$\left\{ \begin{aligned} f_{2i+1,2j}^{k+1} &= \frac{8+\omega}{16}(f_{i,j}^k + f_{i+1,j}^k) - \frac{\omega}{16}(f_{i-1,j}^k + f_{i+2,j}^k), \\ f_{2i,2j+1}^{k+1} &= \frac{8+\omega}{16}(f_{i,j}^k + f_{i,j+1}^k) - \frac{\omega}{16}(f_{i,j-1}^k + f_{i,j+2}^k). \end{aligned} \right.$$

Finally, the face point $f_{2i+1,2j+1}^{k+1}$ is,

$$\begin{aligned}
 f_{2i+1,2j}^{k+1} = & \left(\frac{1}{256} \omega^2 f_{i-1,j-1}^k - \omega(8+\omega) f_{i,j-1}^k - \omega(8+\omega) f_{i+1,j-1}^k \right. \\
 & + \omega^2 f_{i+2,j-1}^k - \omega(8+\omega) f_{i-1,j}^k + (8+\omega)^2 f_{i,j}^k + (8+\omega)^2 f_{i+1,j}^k \\
 & - \omega(8+\omega) f_{i+2,j}^k - \omega(8+\omega) f_{i-1,j+1}^k + (8+\omega)^2 f_{i,j+1}^k \\
 & + (8+\omega)^2 f_{i+1,j+1}^k - \omega(8+\omega) f_{i+2,j+1}^k + \omega^2 f_{i-1,j+2}^k \\
 & \left. - \omega(8+\omega) f_{i,j+2}^k - \omega(8+\omega) f_{i+1,j+2}^k + \omega^2 f_{i+2,j+2}^k \right)
 \end{aligned} \tag{3.15}$$

Instead of taking the tensor product the above rules can be directly obtained by substituting $n = m = 2$ in (3.8) and (3.9), then $\alpha = \beta = 0,1$ and $\gamma, \sigma = 0$, the symmetry conditions are then written as,

$$\begin{cases} a_{-2-\alpha} = a_{2+\alpha}, \\ a_{-2-\beta} = a_{2+\beta}, \end{cases} \tag{3.16}$$

then the formula (3.8) acquires the form,

$$f_{2i+\alpha,2j+\beta}^{k+1} = \sum_{l_1=-1}^2 \sum_{l_2=-1}^2 a_{(2l_1-\sigma,2l_2-\beta)} f_{i+l_1,j+l_2}^k, \tag{3.17}$$

At each mask $a_{i,j}$ of the refinement rule satisfies $a_{i,j} = b_i b_j$, then

$$a_{(2l_1-\sigma,2l_2-\beta)} = a_{2l_1-\alpha} a_{2l_2-\beta}$$

Using the result (3.4) and (3.5) the constants $a_{2l_1-\alpha}$ and $a_{2l_2-\beta}$ are evaluated by substituting $v_1 = v_2 = 0$, $a_3 = -\omega/16$, $\gamma = \sigma = 0$, $s_1, s_2 = 1$ and $a_4 = 0$ for both n and m . After substituting the weights in (3.17) we get the same rules (3.14) and (3.15).

Example: Using the results for the interpolating curve subdivision schemes (2.10) the 4-point interpolatory scheme [3] is obtained. Further here the tensor product of the scheme is evaluated by using the result (3.8). Put $n = m = 3, b_1 = b_2 = 0$ in (3.8),

$$f_{3i+\alpha,3j+\beta}^{k+1} = \sum_{l_1=-1}^2 \sum_{l_2=-1}^2 a_{(3l_1-\sigma,3l_2-\beta)} f_{i+l_1,j+l_2}^k, \tag{3.18}$$

Also, from (3.9)

$$\begin{cases} a_{-2-\alpha} = a_{2+\alpha}, \\ a_{-2-\beta} = a_{2+\beta}, \end{cases}$$

Taking $\alpha = \beta = 0,1,2$, $v_1 = v_2 = a_3 = 0$, $\omega_1 = \omega_2 = a_5 = -5/81$, $\mu_1 = \mu_2 = a_4 = -4/81$. Also for interpolatory scheme $a_{n(\gamma+2)} = a_{m(\sigma+2)} = 0$, as $n = m = 3$ gives $a_6 = 0$ for both n and m .

After calculating the mask from (3.4) and (3.5) and substituting all the results in equation (3.18) following 4-point ternary interpolating tensor product scheme is obtained

$$\begin{aligned}
 f_{3i,3j}^{k+1} &= f_{i,j}^k, \\
 f_{3i+1,3j}^{k+1} &= \frac{1}{81} \left(-5f_{i-1,j}^k + 60f_{i,j}^k + 30f_{i+1,j}^k - 4f_{i+2,j}^k \right), \\
 f_{3i+2,3j}^{k+1} &= \frac{1}{81} \left(-4f_{i-1,j}^k + 30f_{i,j}^k + 60f_{i+1,j}^k - 5f_{i+2,j}^k \right), \\
 f_{3i,3j+1}^{k+1} &= \frac{1}{81} \left(-5f_{i,j-1}^k + 60f_{i,j}^k + 30f_{i,j+1}^k - 4f_{i,j+2}^k \right),
 \end{aligned}$$

$$f_{3i+1,3j+1}^{k+1} = \frac{1}{6561} \left(20f_{i-1,j-1}^k - 240f_{i,j-1}^k - 120f_{i+1,j}^k + 16f_{i+2,j}^k \right. \\ \left. - 150f_{i-1,j}^k + 1800f_{i,j}^k + 900f_{i+1,j}^k - 120f_{i+2,j}^k \right. \\ \left. - 300f_{i-1,j+1}^k + 3600f_{i,j+1}^k + 1800f_{i+1,j+1}^k - 240f_{i+2,j+1}^k \right. \\ \left. + 25f_{i-1,j+2}^k - 300f_{i,j+2}^k - 150f_{i+1,j+2}^k + 20f_{i+2,j+2}^k \right),$$

$$f_{3i+2,3j+1}^{k+1} = \frac{1}{6561} \left(20f_{i-1,j-1}^k - 150f_{i,j-1}^k - 300f_{i+1,j-1}^k + 25f_{i+2,j-1}^k \right. \\ \left. - 240f_{i-1,j}^k + 1800f_{i,j}^k + 3600f_{i+1,j}^k - 300f_{i+2,j}^k \right. \\ \left. - 120f_{i-1,j+1}^k + 900f_{i,j+1}^k + 1800f_{i+1,j+1}^k - 150f_{i+2,j+1}^k \right. \\ \left. + 16f_{i-1,j+2}^k - 120f_{i,j+2}^k - 240f_{i+1,j+2}^k + 20f_{i+2,j+2}^k \right),$$

$$f_{3i,3j+2}^{k+1} = \frac{1}{81} \left(-4f_{i,j-1}^k + 30f_{i,j}^k + 60f_{i,j+1}^k - 5f_{i,j+2}^k \right),$$

$$f_{3i+1,3j+2}^{k+1} = \frac{1}{6561} \left(20f_{i-1,j-1}^k - 240f_{i,j-1}^k - 120f_{i+1,j-1}^k + 16f_{i+2,j-1}^k \right. \\ \left. - 150f_{i-1,j}^k + 1800f_{i,j}^k + 900f_{i+1,j}^k - 120f_{i+2,j}^k \right. \\ \left. - 300f_{i-1,j+1}^k + 3600f_{i,j+1}^k + 1800f_{i+1,j+1}^k - 240f_{i+2,j+1}^k \right. \\ \left. + 25f_{i-1,j+2}^k - 300f_{i,j+2}^k - 150f_{i+1,j+2}^k + 20f_{i+2,j+2}^k \right),$$

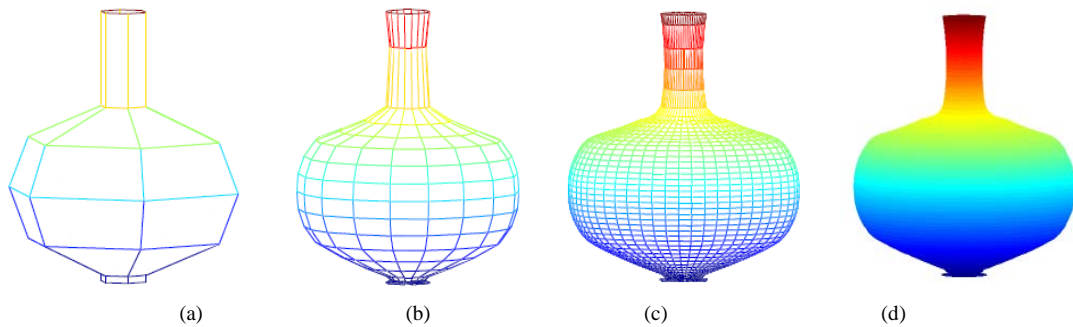


Figure 2. Performance of tensor product of 4-point ternary interpolating scheme: (a), (b), (c) and (d) show the initial polygon, 1st, 2nd subdivision levels and limit surface respectively. (a) Initial polygon; (b) 1st level; (c) 2nd level; (d) Limit surface.

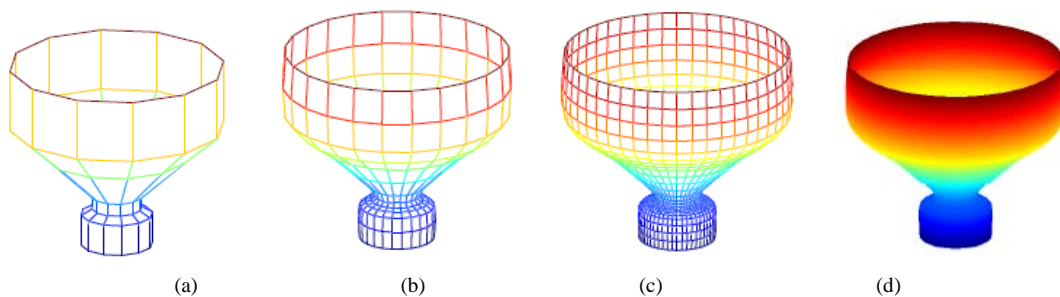


Figure 3. Performance of tensor product of 4-point binary interpolating scheme: (a), (b), (c) and (d) show the initial polygon, 1st, 2nd subdivision levels and limit surface respectively. (a) Initial polygon; (b) 1st level; (c) 2nd level; (d) Lim-

it surface.

$$\begin{aligned}
 f_{3i+2,3j+2}^{k+1} = & \frac{1}{6561} \left(16f_{i-1,j-1}^k - 120f_{i,j-1}^k - 240f_{i+1,j-1}^k + 20f_{i+2,j-1}^k \right. \\
 & - 120f_{i-1,j}^k + 900f_{i,j}^k + 1800f_{i+1,j}^k - 150f_{i+2,j}^k \\
 & - 240f_{i-1,j+1}^k + 1800f_{i,j+1}^k + 3600f_{i+1,j+1}^k - 300f_{i+2,j+1}^k \\
 & \left. + 20f_{i-1,j+2}^k - 150f_{i,j+2}^k - 300f_{i+1,j+2}^k + 25f_{i+2,j+2}^k \right),
 \end{aligned} \tag{3.19}$$

4. Numerical Examples

Here, the performances of some of the schemes which are deduced from the proposed formula are shown. **Figure 2** shows the tensor product of 4-point ternary interpolating scheme (3.19), and **Figure 3** gives the performance of the proposed 4-point binary scheme (3.10).

REFERENCES

- [1] G. Dahlquist and A. Bjork, "Numerical Methods in Scientific Computing," *SIAM*, Vol. 1, 2008. <http://dx.doi.org/10.1137/1.9780898717785>
- [2] L. Kobbelt, "Interpolatory Subdivision on Open Quadrilateral Nets with Arbitrary Topology," *Computer Graphics Forum*, Vol. 15, No. 3, 1996, pp. 409-420. <http://dx.doi.org/10.1111/1467-8659.1530409>
- [3] J. A. Lian, "On A-Ary Subdivision for Curve Design: 4-Point and 6-Point Interpolatory Schemes," *Application and Applied Mathematics: International Journal*, Vol. 3, No. 1, 2008, pp. 18-29.
- [4] J. A. Lian, "On A-Ary Subdivision for Curve Design. II. 3-Point and 5-Point Interpolatory Schemes," *Applications and Applied Mathematics: International Journal*, Vol. 3, No. 2, 2008, pp. 176-187.
- [5] J. A. Lian, "On A-Ary Subdivision for Curve Design. III. 2m-Point and (2m+1)-Point Interpolatory Schemes," *Applications and Applied Mathematics: International Journal*, Vol. 4, No. 2, 2009, pp. 434-444.
- [6] K. P. Ko, "A Study on Subdivision Scheme-Draft," Dongseo University Busan South Korea, 2007. [http://kowon.dongseo.ac.kr/\\$\sim\\$skpko/publication/2004book.pdf](http://kowon.dongseo.ac.kr/\simskpko/publication/2004book.pdf)
- [7] G. Mustafa and A. R. Najma, "The Mask of (2b+4)-Point n-Ary Subdivision Scheme," *Computing*, Vol. 90, No. 1-2, 2010, pp. 1-14.
- [8] M. Sabin, "Eigenanalysis and Artifacts of Subdivision Curves and Surfaces," In: A. Iske, E. Quak and M. S. Floater, Eds., *Tutorials on Multiresolution in Geometric Modelling*, Chapter 4, Springer, Berlin, 2002, pp. 51-68.
- [9] N. A. Dodgson, U. H. Augsdorfer, T. J. Cashman and M. A. Sabin, "Deriving Box-Spline Subdivision Schemes," Springer-Verlag, Berlin, 2009, pp. 106-123.
- [10] N. Dyn, D. Levin and J. Gregory, "A 4-Point Interpolatory Subdivision Scheme for Curve Design," *Computer Aided Geometric Design*, Vol. 4, No. 4, 1987, pp. 257-268. [http://dx.doi.org/10.1016/0167-8396\(87\)90001-X](http://dx.doi.org/10.1016/0167-8396(87)90001-X)