

Modified Method for Controlling, and Generating Design by Using Cubic Bezier Surface in 3D

A. M. J. Abdul-Hussen

Ali Hassan Hammadie

Ali. Adil Saeid

University of Technology, Computer Department

Abstract

The reaction between designer, and design needs modified methods to control the design. This paper presents modified mathematical technique for controlling the generation of the 3D designs of third degree, by using modified Gallier of Bezier surfaces. The paper discusses a bipolynomial surface in terms, of polar forms, first with respect to the parameter, and second with respect to the second parameters, The modified method has resulted in good starting point, to generate which 3D design, algorithm which allows the designer to produce a design in combinational way allows him to get the shape that he has in his mind keeping the 16 control points for 3D design.

The method shows a great flexibility in 3D design controlling, by using the values of the coefficient of parametric representation that uses Three-dimensional cubic Bezier surfaces.

.There is no need to change the control points of the design, moreover efficiency in designs is obtained in comparison with that needed for conventional methods.

الخلاصة

تطوير التفاعل بين المصمم والتصميم يحتاج الى تطوير طرق التحكم بالتصميم وهذا البحث يعرض تقنية رياضية مطورة للسيطرة على توليد التصاميم ثلاثية الأبعاد من الدرجة الثالثة. باستخدام تطوير Gallier لسطح Bezier . تم مناقشة سطح ثنائي متعددة الحدود في الأسقطاب لحدوده المفصلية أولا بالنسبة الى المتغير الأول. وثانيا بالنسبة الى المتغير الثاني. وهذا السبب يعطي نقطة بداية جيدة لأقتراح تقنية لتوليد التصاميم ثلاثية الأبعاد والذي يتيح للمصمم توليد المنحني وتطويره بطريقة تفاعلية تمكّنه من الحصول على الشكل الذي كونه في مخيلته مع الاحتفاظ بتوافقه مع نقاط السيطرة ال (16) الهيكلية للمخطط ثلاثية الأبعاد. أثبتت الطريقة المقترحة مرونة عالية في مجال التحكم بالتصميم وذلك باستخدام معاملات المتغيرات في سطح Bezier من دون تغيير نقاط السيطرة للتصميم. كذلك أثبتنا الطريقة أنها أكثر كفاءة في التصاميم بالمقارنة مع ما يوجد من طرق سابقة.

Introduction

The 3D design is an important method in modern technology because there are many in daily life applications would be needed in everyday life.

This paper takes up the study of a bipolynomial surface. Bipolynomial surface is defined in terms, of polar forms. Because the bipolynomial surface involved contains two variables, natural way to polarize polynomial surface. The approach yields bipolynomial surface {also called tensor product surfaces}. It is shown versions of the de-Casteljau algorithm can be turned into subdivisions, by giving an efficient method of performing subdivision. It is also shown that it is easy to compute a new control net from given net. Intuitively this depends on the parameters; this is one of indications that deals with surface. This is far more complex than dealing with curves. The affine frame, for simplicity of notation is denoted as $F(u, v)$. Intuitively, a polynomial surface is obtained by bending and twisting the real affine plane using a polynomial map. This present method a differen method, for controlling and generating the 3D design. The arithmetical technique is used to generate Gallier cubic Bezier surfaces by using de-Casteljau algorithm (Bezier curves are named after Bezier for his work at Renault in the 1960s. Slightly earlier de-Casteljau had already developed mathematically equivalent method of defining Bezier curve). This new work is modified for controlling and generatn surface, by using the values of the coefficient of parametric surfaces representation that uses Three-dimensional cubic Bezier surfaces, with no need to change the control points of Bezier surface.

Note I [Gallier 00], [Hill 2001], [Jaber 05].

(i)-A linear combination of m vectors $V_1, V_2, \dots, V_m$ is a vector, say W , of the form

$$W = a_1 V_1 + a_2 V_2 + \dots + a_m V_m, \dots (*)$$

where $a_1, a_2, \dots, a_m$ are scalar.

A linear combination of vectors is called an affine combination if the coefficients $a_1, a_2, \dots, a_m$ are added up to unity. Thus the linear combination in Eq (*) is affine if

$$a_1 + a_2 + \dots + a_m = 1.$$

The coefficients of an affine combination of two vectors V_1 and V_2 are often forced to have scalar t for one of them and $(1-t)$ for the other in the form:

$$W = (1-t)V_1 + tV_2$$

The linear combination of Eq (*) is convex if it is satisfying the following two conditions: -

(1)- $a_1 + a_2 + \dots + a_m = 1$, (2)- $a_i \geq 0$ for $i=1 \dots m$, as a consequence all a_i must lie between 0 and 1.

(ii)- Affine space A^n is called the real affine space of dimension n . In most case, we will consider, ($n=1, 2, 3$). Typically, the 1D affine line A^1 , or the 2D affine plane A^2 and 3D affine space A^3 , A is defined over the field R (where R is real numbers).

Polynomial Curve in Polar Form (2D). [Gallier 00] [Jaber 05]

A method of specifying polynomial curves (2D) that yields very nice geometric construction of the curves is based on polar polynomial form. To define the polar form of a polynomial of degree three, let

$$X(u) = F_1(u) = a_1 u^3 + b_1 u^2 + c_1 u + d_1 \quad (1)$$

$$Y(u) = F_2(u) = a_2 u^3 + b_2 u^2 + c_2 u + d_2, \quad (2)$$

where $a_1, b_1, c_1, d_1, a_2, b_2, c_2$ and d_2 are constants. The polar form of F is a symmetric affine function

$f: A^3 \rightarrow A$ that takes the same value for all permutations of u_1, u_2, u_3 . That is

$$f(u_1, u_2, u_3) = f(u_2, u_1, u_3) = f(u_1, u_3, u_2) = f(u_2, u_3, u_1) = f(u_3, u_1, u_2) = f(u_3, u_2, u_1).$$

Which is affine in each argument and such that $F(u) = f(u, u, u)$, for all u belong to R . It is easily verified that F must be given by

$$f(u_1, u_2, u_3) = a u_1 u_2 u_3 + b [u_1 u_2 + u_1 u_3 + u_2 u_3]/3 + c [u_1 + u_2 + u_3]/3 + d \quad (3)$$

This is called the polar form ($\oplus$) of cubic polynomial in (1) or (2) [Jaber 05] [Gallier 00]

Example1:

Consider the plane cubic defined as follows

$$F_1(u) = 15u \quad (4)$$

$$F_2(u) = 9u - u^3 \quad (5)$$

The polar forms

$$f_1(u_1, u_2, u_3) = 3[u_1 + u_2 + u_3] \quad (6)$$

$$f_2(u_1, u_2, u_3) = 3[u_1 + u_2 + u_3] - u_1 u_2 u_3. \quad (7)$$

It is easy to notice that $f_1(u_1, u_2, u_3)$ in (6) is polar form of F_1 in (4), and $f_2(u_1, u_2, u_3)$ in (7) is polar form of F_2 in (5).

$\oplus$ Polar form new definition has no relation to polar coordinate.

Bipolynomial Surface in Polar Form . (3D). [Gallier 00].

The treatment of parametric cubic curve segments given in foregoing section is easily generalized to bi-parametric in a bipolynomial surface, (3D). A point on the surface is given by bi-parametric function and a set of blending or basis function is used for each parameter.

Now the study investigates the possibility to define bipolynomial surface in terms, of polar forms. Given a bipolynomial surface, (3D) of degree $\langle 3, 3 \rangle$, each polynomial F_i

(u, v) defined F , first with respect to u , and then with respect to v to get polar forms. Note that they are intentionally denoting

$\underbrace{A \times A \times A}_3 \times \underbrace{A \times A \times A}_3$, instead of $A^3 \times A^3$, to avoid the confusion between the affine space A^3 and the Cartesian product $\underbrace{A \times A \times A}_3$.

The advantage of this method is that it allows the use of many algorithms applied in the case of curves. Note that they in this case

$$F(u, v) = f(\underbrace{u \times u \times u}_3 \times \underbrace{v \times v \times v}_3),$$

The present method investigates appropriate generalizations of de-Casteljau algorithm, consider the following example to illustrate the way to polarize separately u and v . Using linearity, it is enough to explain how to polarize a monomial $F(u, v)$ of the forme $u^3 v^3$ with respect to the bidegree $\langle 3, 3 \rangle$.

In order to find the polar form $f_1(u, u, u; v, v, v)$ of F viewed as a bi polynomial surface of degree $\langle 3, 3 \rangle$, polarize each of the $F_i(u, v)$ separately in u and v .

It is quite obvious that the same result is obtained if you first polarize with respect to u , and then with respect to v , or conversely. It is easily seen that, as in example 2

Example2:

Consider the following surface viewed as a bipolynomial surface of degree $\langle 3, 3 \rangle$:

$$X(u, v) = F_1(u, v) = u - \frac{u^3}{3} + uv^2 \quad (8)$$

$$Y(u, v) = F_2(u, v) = v - \frac{v^3}{3} + vu^2 \quad (9)$$

$$Z(u, v) = F_3(u, v) = u^2 - v^2 \quad (10)$$

Let us now polarize surface of total degree $\langle 3, 3 \rangle$ in A^3 . After polarizing with respect to u you have:

$$F_1U(u_1, u_2, u_3, v) = \frac{u_1 + u_2 + u_3}{3} - \frac{u_1 u_2 u_3}{3} + \frac{u_1 + u_2 + u_3}{3} v^2$$

$$F_2U(u_1, u_2, u_3, v) = v - \frac{v^3}{3} + v \frac{u_1 u_2 + u_1 u_3 + u_2 u_3}{3}$$

$$F_3U(u_1, u_2, u_3, v) = \frac{u_1 u_2 + u_1 u_3 + u_2 u_3}{3} - v^2,$$

And polarizing with respect to v yield

$$F_1(u_1, u_2, u_3; v_1, v_2, v_3) = \frac{u_1 + u_2 + u_3}{3} - \frac{u_1 u_2 u_3}{3} + \frac{u_1 + u_2 + u_3}{3} \frac{v_1 v_2 + v_1 v_3 + v_2 v_3}{3} \quad (11)$$

$$F_2(u_1, u_2, u_3; v_1, v_2, v_3) = \left[\frac{v_1 + v_2 + v_3}{3} \right] - \frac{v_1 v_2 v_3}{3} + \left[\frac{v_1 + v_2 + v_3}{3} \right] \left[\frac{u_1 u_2 + u_1 u_3 + u_2 u_3}{3} \right] + \frac{u_1 u_2 + u_1 u_3 + u_2 u_3}{3} \quad (12)$$

$$F_3(u_1, u_2, u_3; v_1, v_2, v_3) = \frac{u_1 u_2 + u_1 u_3 + u_2 u_3}{3} - \frac{v_1 v_2 + v_1 v_3 + v_2 v_3}{3}. \quad (13)$$

It is easy to notice that Eqs (11, 12 and 13) are polar forms of Eqs (8, 9 and 10), respectively.

Background

De-Casteljau algorithm is an algorithm which uses a sequence of control points, P_1, P_2, P_3, P_4 to construct a well defined curve $F(u)$ at each value of u from 0 to 1. This

provides a way to generate a curve from a set of points. Changing the points will change the curve. $F(u)$ defined as: [Faux 83], [Buss03], [Lengyel 04], [Jaber 10] [Jaber 05]

$$P(u) = (1-u)^3 P_1 + 3(1-u)^2 u P_2 + 3(1-u) u^2 P_3 + u^3 P_4. \quad (14)$$

Eq (14) called original cubic Bezier curve (2D) is dependent on interval $[0, 1]$, and uses a sequence of control points $P_1 \dots P_4$ to define 3D surfaces (Mathematically are said to be generated from the Cartesian product of two curves). A cubic Bezier surfaces is defined as: [Hill 01] [Gerald 99]. [Faux 83], [Watt 00], [Lengyel 04], [Klawonn 08].

$$F(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 \binom{3}{i} \binom{3}{j} [1-u]^{3-i} [u]^i [1-v]^{3-j} [v]^j P_{ij} \quad (15)$$

Eq (15) is called original cubic Bezier surfaces.

Gallier Modified De- Casteljau Algorithmic. [Gallier 00], [Jaber 05].

The powerful method of subdivision is also discussed extensively. It is written in mathematical form to illustrate concretely the de Casteljau algorithm at cubic curve. The modified method uses a sequence of control points $p_i = f(r^{3-i} s^i)$, $\{(i=0, 1, 2, 3)\}$ $f(r^3) = f(r, r, r)$, $f(r^2 s) = f(r, r, s)$, $f(r s^2) = f(r, s, s)$ and $f(s^3) = f(s, s, s)$ to construct a well defined curve $F(u) = f(u, u, u)$. Let us begin with straight lines. Given affine frame $[r, s]$ which is $r < s$, $u \in [r, s]$, $u \in R$. Let us begin with straight lines (degree one), which can be written uniquely as

$u = [1-\lambda] r + \lambda s = r + \lambda [s-r]$. And you can find that:

$$\lambda = \frac{u-r}{s-r}, \text{ and } 1-\lambda = \frac{s-u}{s-r}. \text{ Since } F \text{ is affine then}$$

$$F(u) = F([1-\lambda] r + \lambda s) = (1-\lambda) F(r) + \lambda F(s). \quad (16)$$

De Castejalu algorithm uses two control points say $F(r)$ and $F(s)$ as in Eq (16). Eq (16) called Gallier modified Bezier curve is dependent on interval $[r, s]$.

De Castejalu algorithm at cubic curve (Degree three) is

$$f(u, u, u) = f(u, u, (1-\lambda)r + \lambda s) = (1-\lambda) f(u, u, r) + \lambda f(u, u, s), \quad (17)$$

in terms of $f(u, u, r)$ and $f(u, u, s)$. From $f(u, u, r)$, $f(u, u, s)$ are given by:

$$f(u, u, r) = f(u, (1-\lambda)r + \lambda s, r) = (1-\lambda) f(u, r, r) + \lambda f(u, s, r) \quad (18)$$

$$f(u, u, s) = f(u, (1-\lambda)r + \lambda s, s) = (1-\lambda) f(u, r, s) + \lambda f(u, s, s) \quad (19)$$

From $f(u, r, r)$, $f(u, r, s)$, $f(u, s, s)$ are given by:

$$f(r, r, u) = f(r, r, (1-\lambda)r + \lambda s) = (1-\lambda) f(r, r, r) + \lambda f(r, r, s) \quad (20)$$

$$f(r, u, s) = f(r, (1-\lambda)r + \lambda s, s) = (1-\lambda) f(r, r, s) + \lambda f(r, s, s) \quad (21)$$

$$f(u, s, s) = f((1-\lambda)r + \lambda s, s, s) = (1-\lambda) f(r, s, s) + \lambda f(s, s, s). \quad (22)$$

From symmetry, $f(r, r, u) = f(u, r, r)$ and $f(r, u, s) = f(u, s, r) = f(u, r, s)$, and substitution of Eqs (18, 19, 20, 21) and (22) in (17) gives

$$f(u, u, u) = (1-\lambda)^3 f(r, r, r) + 3\lambda(1-\lambda)^2 f(r, r, s) + 3(1-\lambda)\lambda^2 f(r, s, s) + \lambda^3 f(s, s, s). \quad (23)$$

Using de Casteljau algorithm in cubic cases, defines the curve $F(u)$ at each value of u from r to s , suppose $f(r, r, r) = F(u)$, and $f(r, r, s)$, $f(r, s, s)$, $f(s, s, s) = P_0, P_1, P_2, P_3$ are control points Eq (23) becomes

$$F(u) = (1-\lambda)^3 p_0 + 3\lambda(1-\lambda)^2 p_1 + 3(1-\lambda)\lambda^2 p_2 + \lambda^3 p_3. \quad (24)$$

Eq (24) called original Gallier modified Bezier curve is dependent on interval $[r, s]$ (2D). [Gallier 00], [Jaber 05].

These control points play a major role in the de-Casteljau algorithm and its extensions. The polynomial curve defining F passes through the two points p_0 and p_3 , but not through the other control points. For $r=0$ and $s=1$, Eq (24) becomes: -

$$F(u) = (1-u)^3 P_1 + 3(1-u)^2 u P_2 + 3(1-u) u^2 P_3 + u^3 P_4. \quad (25)$$

Eq (25) called original Bezier curve is dependent on frame $[0, 1]$. It is identical to Eq (14). Now the de-Casteljau algorithm can be discussed in some detail in bipolynomial surfaces. De-Casteljau algorithm can be generalized very easily to bipolynomial surfaces. Using linearity, it is enough to deal with 3D, let $[r_1, s_1]$ and $[r_2, s_2]$ be two affine frames for the affine line A . Every point $u \in R$ can be written as

$u = [1 - \lambda] r_1 + \lambda s_1 = r_1 + \lambda [s_1 - r_1]$. And you can find that:

$\lambda = \frac{u - r_1}{s_1 - r_1}$, and $1 - \lambda = \frac{s_1 - u}{s_1 - r_1}$. Similarly any point $v \in A$ can be written as

$v = [1 - \beta] r_2 + \beta s_2 = r_2 + \beta [s_2 - r_2]$. And you can find that:

$\beta = \frac{v - r_2}{s_2 - r_2}$, and $1 - \beta = \frac{s_2 - v}{s_2 - r_2}$.

The treatment of parametric cubic curve segments given in foregoing paragraphs is easily generalized to bi-parametric cubic surface. A point on the surface is given by bi-parametric function and a set of blending or basis function is used for each parameter.

Now de-Casteljau algorithm at bi-cubic (3D) is defined as:

$$\begin{aligned} f_1(u_1, u_2, u_3; v_1, v_2, v_3) &= [(1-\lambda_1)(1-\beta_1)] f(r_1, u_2, u_3; r_2, v_2, v_3) \\ &+ [(1-\lambda_1)\beta_1] f(r_1, u_2, u_3; s_2, v_2, v_3) + [\lambda_1(1-\beta_1)] f(s_1, u_2, u_3; r_2, v_2, v_3) \\ &+ [\lambda_1\beta_1] f(s_1, u_2, u_3; s_2, v_2, v_3). \end{aligned} \quad (26)$$

Let $a = [(1-\lambda_1)(1-\beta_1)] f(r_1, u_2, u_3; r_2, v_2, v_3)$, $b = [(1-\lambda_1)\beta_1] f(r_1, u_2, u_3; s_2, v_2, v_3)$, $c = [\lambda_1(1-\beta_1)] f(s_1, u_2, u_3; r_2, v_2, v_3)$ $d = [\lambda_1\beta_1] f(s_1, u_2, u_3; s_2, v_2, v_3)$.

$\therefore$ Eq (26) becomes

$$f_1(u_1, u_2, u_3; v_1, v_2, v_3) = a + b + c + d. \quad (27)$$

From a:

$$\begin{aligned} f(r_1, u_2, u_3; r_2, v_2, v_3) &= (1-\lambda_2)(1-\beta_2) f(r_1, r_1, u_3; r_2, r_2, v_3) \\ &+ (1-\lambda_2)\beta_2 f(r_1, r_1, u_3; r_2, s_2, v_3) + \lambda_2(1-\beta_2) f(r_1, s_1, u_3; r_2, r_2, v_3) \\ &+ \lambda_2\beta_2 f(r_1, s_1, u_3; r_2, s_2, v_3) \end{aligned} \quad (28)$$

$$\begin{aligned} f(r_1, r_1, u_3; r_2, r_2, v_3) &= (1-\lambda_3)(1-\beta_3) f(r_1, r_1, r_1; r_2, r_2, r_2) \\ &+ (1-\lambda_3)\beta_3 f(r_1, r_1, r_1; r_2, r_2, s_2) + \lambda_3(1-\beta_3) f(r_1, r_1, s_1; r_2, r_2, r_2) \\ &+ \lambda_3\beta_3 f(r_1, r_1, s_1; r_2, r_2, s_2) \end{aligned} \quad (29)$$

$$\begin{aligned} f(r_1, r_1, u_3; r_2, s_2, v_3) &= (1-\lambda_3)(1-\beta_3) f(r_1, r_1, r_1; r_2, s_2, r_2) \\ &+ (1-\lambda_3)\beta_3 f(r_1, r_1, r_1; r_2, s_2, s_2) + \lambda_3(1-\beta_3) f(r_1, r_1, s_1; r_2, s_2, r_2) \\ &+ \lambda_3\beta_3 f(r_1, r_1, s_1; r_2, s_2, s_2) \end{aligned} \quad (30)$$

$$\begin{aligned} f(r_1, s_1, u_3; r_2, r_2, v_3) &= (1-\lambda_3)(1-\beta_3) f(r_1, s_1, r_1; r_2, r_2, r_2) \\ &+ (1-\lambda_3)\beta_3 f(r_1, s_1, r_1; r_2, r_2, s_2) + \lambda_3(1-\beta_3) f(r_1, s_1, s_1; r_2, r_2, r_2) \\ &+ \lambda_3\beta_3 f(r_1, s_1, s_1; r_2, r_2, s_2) \end{aligned} \quad (31)$$

$$\begin{aligned} f(r_1, s_1, u_3; r_2, s_2, v_3) &= (1-\lambda_3)(1-\beta_3) f(r_1, s_1, r_1; r_2, s_2, r_2) \\ &+ (1-\lambda_3)\beta_3 f(r_1, s_1, r_1; r_2, s_2, s_2) + \lambda_3(1-\beta_3) f(r_1, s_1, s_1; r_2, s_2, r_2) \\ &+ \lambda_3\beta_3 f(r_1, s_1, s_1; r_2, s_2, s_2) \end{aligned} \quad (32)$$

From symmetry, $f(r_1, r_1, s_1; r_2, r_2, r_2) = f(r_1, s_1, r_1; r_2, r_2, r_2)$ and $f(r_1, r_1, s_1; r_2, r_2, s_2) = f(r_1, r_1, s_1; r_2, s_2, r_2) = f(r_1, s_1, r_1; r_2, r_2, s_2)$, and substitution of equations (29, 30, 31,) and (32) in (28) gives:

$$\begin{aligned} f(r_1, u_2, u_3; r_2, v_2, v_3) &= (1-\lambda_2)(1-\beta_2)(1-\lambda_3)(1-\beta_3) f(r_1, r_1, r_1; r_2, r_2, r_2) \\ &+ (1-\lambda_2)(1-\beta_2)(1-\lambda_3)\beta_3 f(r_1, r_1, r_1; r_2, r_2, s_2) + (1-\lambda_2)(1-\beta_2)\lambda_3(1-\beta_3) f(r_1, r_1, s_1; r_2, r_2, r_2) \\ &+ (1-\lambda_2)(1-\beta_2)\lambda_3\beta_3 f(r_1, r_1, s_1; r_2, r_2, s_2) + (1-\lambda_2)\beta_2(1-\lambda_3)(1-\beta_3) f(r_1, r_1, r_1; r_2, s_2, r_2) \\ &+ (1-\lambda_2)\beta_2(1-\lambda_3)\beta_3 f(r_1, r_1, r_1; r_2, s_2, s_2) + (1-\lambda_2)\beta_2\lambda_3(1-\beta_3) f(r_1, r_1, s_1; r_2, s_2, r_2) \\ &+ (1-\lambda_2)\beta_2\lambda_3\beta_3 f(r_1, r_1, s_1; r_2, s_2, s_2) + \lambda_2(1-\beta_2)(1-\lambda_3)(1-\beta_3) f(r_1, s_1, r_1; r_2, r_2, r_2) \\ &+ \lambda_2(1-\beta_2)(1-\lambda_3)\beta_3 f(r_1, s_1, r_1; r_2, r_2, s_2) + \lambda_2(1-\beta_2)\lambda_3(1-\beta_3) f(r_1, s_1, s_1; r_2, r_2, r_2) \\ &+ \lambda_2(1-\beta_2)\lambda_3\beta_3 f(r_1, s_1, s_1; r_2, r_2, s_2) + \lambda_2\beta_2(1-\lambda_3)(1-\beta_3) f(r_1, s_1, r_1; r_2, s_2, r_2) \\ &+ \lambda_2\beta_2(1-\lambda_3)\beta_3 f(r_1, s_1, r_1; r_2, s_2, s_2) + \lambda_2\beta_2\lambda_3(1-\beta_3) f(r_1, s_1, s_1; r_2, s_2, r_2) \\ &+ \lambda_2\beta_2\lambda_3\beta_3 f(r_1, s_1, s_1; r_2, s_2, s_2). \end{aligned} \quad (33)$$

Now substituting Eq (33) in (a) gives:-

$$\begin{aligned} a &= [(1-\lambda_1)(1-\beta_1)] f(r_1, u_2, u_3; r_2, v_2, v_3) = (1-\lambda_1)(1-\beta_1)(1-\lambda_2)(1-\beta_2)(1-\lambda_3)(1-\beta_3) f \\ &(r_1, r_1, r_1; r_2, r_2, r_2) + (1-\lambda_1)(1-\beta_1)(1-\lambda_2)(1-\beta_2)(1-\lambda_3)\beta_3 f(r_1, r_1, r_1; r_2, r_2, s_2) + (1-\lambda_1)(1-\beta_1)(1-\lambda_2)(1-\beta_2)\lambda_3(1-\beta_3) f(r_1, r_1, s_1; r_2, r_2, r_2) \\ &+ (1-\lambda_1)(1-\beta_1)(1-\lambda_2)(1-\beta_2)\lambda_3\beta_3 f(r_1, r_1, s_1; r_2, r_2, s_2) + (1-\lambda_1)(1-\beta_1)(1-\lambda_2)\beta_2(1-\lambda_3)(1-\beta_3) f(r_1, r_1, r_1; r_2, s_2, r_2) \\ &+ (1-\lambda_1)(1-\beta_1)(1-\lambda_2)\beta_2(1-\lambda_3)\beta_3 f(r_1, r_1, r_1; r_2, s_2, s_2) + (1-\lambda_1)(1-\beta_1)(1-\lambda_2)\beta_2\lambda_3(1-\beta_3) f(r_1, r_1, s_1; r_2, s_2, r_2) \\ &+ (1-\lambda_1)(1-\beta_1)(1-\lambda_2)\beta_2\lambda_3\beta_3 f(r_1, r_1, s_1; r_2, s_2, s_2) + \lambda_2(1-\beta_1)(1-\lambda_1)(1-\beta_2)(1-\lambda_3)(1-\beta_3) f(r_1, s_1, r_1; r_2, r_2, r_2) \\ &+ \lambda_2(1-\beta_1)(1-\lambda_1)(1-\beta_2)(1-\lambda_3)\beta_3 f(r_1, s_1, r_1; r_2, r_2, s_2) + \lambda_2(1-\beta_1)(1-\lambda_1)\lambda_3(1-\beta_3) f(r_1, s_1, s_1; r_2, r_2, r_2) \\ &+ \lambda_2(1-\beta_1)(1-\lambda_1)\lambda_3\beta_3 f(r_1, s_1, s_1; r_2, r_2, s_2) + \lambda_2\beta_1(1-\lambda_1)(1-\beta_2)(1-\lambda_3)(1-\beta_3) f(r_1, s_1, r_1; r_2, s_2, r_2) \\ &+ \lambda_2\beta_1(1-\lambda_1)(1-\beta_2)(1-\lambda_3)\beta_3 f(r_1, s_1, r_1; r_2, s_2, s_2) + \lambda_2\beta_1\lambda_3(1-\beta_1)(1-\beta_2)(1-\beta_3) f(r_1, s_1, s_1; r_2, s_2, r_2) \\ &+ \lambda_2\beta_1\lambda_3\beta_3 f(r_1, s_1, s_1; r_2, s_2, s_2). \end{aligned}$$

$$\begin{aligned}
& (1-\beta_2) \lambda_3 \beta_3 f(r_1, r_1, s_1; r_2, r_2, s_2) + (1-\lambda_1) (1-\beta_1) (1-\lambda_2) \beta_2 (1-\lambda_3) (1-\beta_3) f(r_1, r_1, r_1; r_2, \\
& s_2, r_2) + (1-\lambda_1) (1-\beta_1) (1-\lambda_2) \beta_2 (1-\lambda_3) \beta_3 f(r_1, r_1, r_1; r_2, s_2, s_2) + (1-\lambda_1) (1-\beta_1) \\
& (1-\lambda_2) \beta_2 \lambda_3 (1-\beta_3) f(r_1, r_1, s_1; r_2, s_2, r_2) + (1-\lambda_1) (1-\beta_1) (1-\lambda_2) \beta_2 \lambda_3 \beta_3 f(r_1, r_1, s_1; r_2, \\
& s_2, s_2) + (1-\lambda_1) (1-\beta_1) \lambda_2 (1-\beta_2) (1-\lambda_3) (1-\beta_3) f(r_1, s_1, r_1; r_2, r_2, r_2) + (1-\lambda_1) \\
& (1-\beta_1) \lambda_2 (1-\beta_2) (1-\lambda_3) \beta_3 f(r_1, s_1, r_1; r_2, r_2, s_2) + (1-\lambda_1) (1-\beta_1) \lambda_2 (1-\beta_2) \lambda_3 (1-\beta_3) f \\
& (r_1, s_1, s_1; r_2, r_2, r_2) + (1-\lambda_1) (1-\beta_1) \lambda_2 (1-\beta_2) \lambda_3 \beta_3 f(r_1, s_1, s_1; r_2, r_2, s_2) + (1-\lambda_1) (1-\beta_1) \\
& \lambda_2 \beta_2 (1-\lambda_3) (1-\beta_3) f(r_1, s_1, r_1; r_2, s_2, r_2) + (1-\lambda_1) (1-\beta_1) \lambda_2 \beta_2 (1-\lambda_3) \beta_3 f(r_1, s_1, r_1; r_2, \\
& s_2, s_2) + (1-\lambda_1) (1-\beta_1) \lambda_2 \beta_2 \lambda_3 (1-\beta_3) f(r_1, s_1, s_1; r_2, s_2, r_2) + (1-\lambda_1) (1-\beta_1) \lambda_2 \beta_2 \lambda_3 \beta_3 f(r_1, \\
& s_1, s_1; r_2, s_2, s_2)
\end{aligned}$$

In same way b, c and d, can be found by using symmetry of the following points [Gallier 00], and out them as

$$\begin{aligned}
\{ & f(r_1, r_1, r_1; r_2, s_2, r_2) = f(r_1, r_1, r_1; s_2, r_2, r_2) = f(r_1, r_1, r_1; r_2, r_2, s_2), f(r_1, r_1, r_1; s_2, \\
& r_2, s_2) = f(r_1, r_1, r_1; s_2, s_2, r_2) = f(r_1, r_1, r_1; r_2, s_2, s_2), f(r_1, s_1, r_1; r_2, r_2, r_2) = f(s_1, r_1, r_1 \\
& ; r_2, r_2, r_2) = f(r_1, r_1, s_1; r_2, r_2, r_2), f(r_1, r_1, s_1; r_2, s_2, r_2) = f(r_1, s_1, r_1; r_2, r_2, s_2) = f(r_1, \\
& s_1, r_1; r_2, s_2, r_2) = f(r_1, s_1, r_1; s_2, r_2, r_2) = f(s_1, r_1, r_1; r_2, r_2, s_2) = f(s_1, r_1, r_1; r_2, s_2, r_2) \\
&) = f(s_1, r_1, r_1; s_2, r_2, r_2) = f(r_1, r_1, s_1; s_2, r_2, r_2) = f(r_1, r_1, s_1; r_2, r_2, s_2), f(r_1, s_1, r_1; r_2, \\
& s_2, s_2) = f(r_1, r_1, s_1; s_2, r_2, s_2) = f(r_1, r_1, s_1; s_2, s_2, r_2) = f(r_1, s_1, r_1; s_2, r_2, s_2) = f(r_1, s_1, \\
& r_1; s_2, s_2, r_2) = f(s_1, r_1, r_1; r_2, s_2, s_2) = f(s_1, r_1, r_1; s_2, r_2, s_2) = f(s_1, r_1, r_1; s_2, s_2, r_2) = f \\
& (r_1, r_1, s_1; r_2, s_2, s_2), f(r_1, s_1, r_1; s_2, s_2, s_2) = f(s_1, r_1, r_1; s_2, s_2, s_2) = f(r_1, r_1, s_1; s_2, \\
& s_2, s_2), f(s_1, r_1, s_1; r_2, r_2, r_2) = f(s_1, s_1, r_1; r_2, r_2, r_2) = f(r_1, s_1, s_1; r_2, r_2, r_2), f(r_1, s_1, s_1 \\
& ; r_2, r_2, s_2) = f(r_1, s_1, s_1; r_2, s_2, r_2) = f(s_1, r_1, s_1; r_2, r_2, s_2) = f(s_1, r_1, s_1; r_2, s_2, r_2) = f \\
& (s_1, s_1, r_1; r_2, r_2, s_2) = f(s_1, s_1, r_1; r_2, s_2, r_2) = f(s_1, r_1, s_1; s_2, r_2, r_2) = f(s_1, s_1, r_1; s_2, r_2, \\
& r_2) = f(r_1, s_1, s_1; s_2, r_2, r_2), f(r_1, s_1, s_1; s_2, r_2, s_2) = f(r_1, s_1, s_1; s_2, s_2, r_2) = f(s_1, r_1, s_1 \\
& ; r_2, s_2, s_2) = f(s_1, s_1, r_1; r_2, s_2, s_2) = f(s_1, r_1, s_1; s_2, r_2, s_2) = f(s_1, r_1, s_1; s_2, s_2, r_2) = f \\
& (s_1, s_1, r_1; s_2, r_2, s_2) = f(s_1, s_1, r_1; s_2, s_2, r_2)] = f(r_1, s_1, s_1; r_2, s_2, s_2), f(s_1, s_1, r_1; s_2, s_2, \\
& s_2) = f(s_1, r_1, s_1; s_2, s_2, s_2) = f(r_1, s_1, s_1; s_2, s_2, s_2), f(s_1, s_1, s_1; r_2, r_2, r_2), f(s_1, s_1, s_1; s_2 \\
& , r_2, r_2) = f(s_1, s_1, s_1; r_2, s_2, r_2)] = f(s_1, s_1, s_1; r_2, r_2, s_2), f(s_1, s_1, s_1; s_2, s_2, r_2) = f(s_1, s_1, \\
& s_1; s_2, r_2, s_2) = f(s_1, s_1, s_1; r_2, s_2, s_2) \}
\end{aligned}$$

From symmetry, the above points when substituted in Eq (27) give

$$\begin{aligned}
\therefore f_1(u_1, u_2, u_3; v_1, v_2, v_3) &= a + b + c + d = \{ (1-\lambda_1) (1-\lambda_2) (1-\lambda_3) (1-\beta_1) (1-\beta_2) (1-\beta_3) f \\
& (r_1, r_1, r_1; r_2, r_2, r_2) + [(1-\lambda_1)(1-\lambda_2) (1-\lambda_3) (1-\beta_1)(1-\beta_2) \beta_3 + (1-\lambda_1)(1-\lambda_2) (1-\lambda_3) (1-\beta_1)\beta_2 \\
& (1-\beta_3) + (1-\lambda_1)(1-\lambda_2) (1-\lambda_3) \beta_1(1-\beta_2) (1-\beta_3)] f(r_1, r_1, r_1; r_2, r_2, s_2) + [(1-\lambda_1)(1-\lambda_2) (1-\lambda_3) \\
& (1-\beta_1)\beta_2\beta_3 + (1-\lambda_1)(1-\lambda_2) (1-\lambda_3) \beta_1(1-\beta_2) \beta_3 + (1-\lambda_1)(1-\lambda_2) (1-\lambda_3) \beta_1\beta_2(1-\beta_3)] f(r_1, r_1, \\
& r_1; r_2, s_2, s_2) + [(1-\lambda_1)(1-\lambda_2) (1-\lambda_3) \beta_1\beta_2\beta_3] f(r_1, r_1, r_1; s_2, s_2, s_2) + [(1-\lambda_1) (1-\lambda_2) \lambda_3 \\
& (1-\beta_1) (1-\beta_2) (1-\beta_3) + (1-\lambda_1)\lambda_2(1-\lambda_3) (1-\beta_1)(1-\beta_2) (1-\beta_3) + \lambda_1(1-\lambda_2) (1-\lambda_3) (1-\beta_1)(1-\beta_2) \\
& (1-\beta_3)] f(r_1, r_1, s_1; r_2, r_2, r_2) + [(1-\lambda_1)(1-\lambda_2) \lambda_3 (1-\beta_1)(1-\beta_2) \beta_3 + (1-\lambda_1)\lambda_2 (1-\lambda_3) (1-\beta_1)(1-\beta_2) \beta_3 \\
& + \lambda_1(1-\lambda_2) (1-\lambda_3) (1-\beta_1)(1-\beta_2) \beta_3 + (1-\lambda_1)(1-\lambda_2) \lambda_3(1-\beta_1)\beta_2 (1-\beta_3) + (1-\lambda_1)\lambda_2 \\
& (1-\lambda_3) (1-\beta_1)\beta_2 (1-\beta_3) + \lambda_1(1-\lambda_2) (1-\lambda_3) (1-\beta_1) \beta_2 (1-\beta_3) + (1-\lambda_1)(1-\lambda_2) \lambda_3 \beta_1(1-\beta_2) (1-\beta_3) \\
& + (1-\lambda_1)\lambda_2 (1-\lambda_3) \beta_1(1-\beta_2) (1-\beta_3) + \lambda_1(1-\lambda_2) (1-\lambda_3) \beta_1(1-\beta_2) (1-\beta_3)] f(r_1, r_1, s_1; r_2, r_2, s_2) \\
& + [(1-\lambda_1)(1-\lambda_2) \lambda_3 (1-\beta_1) \beta_2 \beta_3 + (1-\lambda_1)\lambda_2 (1-\lambda_3) (1-\beta_1) \beta_2 \beta_3 + \lambda_1(1-\lambda_2) (1-\lambda_3) (1-\beta_1)\beta_2\beta_3 \\
& + (1-\lambda_1)(1-\lambda_2) \lambda_3 \beta_1(1-\beta_2) \beta_3 + (1-\lambda_1)\lambda_2 (1-\lambda_3) \beta_1(1-\beta_2) \beta_3 + \lambda_1(1-\lambda_2) (1-\lambda_3) \beta_1(1-\beta_2) \beta_3 \\
& + (1-\lambda_1)(1-\lambda_2) \lambda_3 \beta_1 \beta_2 (1-\beta_3) + (1-\lambda_1)\lambda_2 (1-\lambda_3) \beta_1 \beta_2 (1-\beta_3) + \lambda_1(1-\lambda_2) (1-\lambda_3) \beta_1 \beta_2(1-\beta_3)] \\
& f(r_1, r_1, s_1; r_2, s_2, s_2) + [(1-\lambda_1) (1-\lambda_2) \lambda_3 \beta_1\beta_2 \beta_3 + (1-\lambda_1) \lambda_2 (1-\lambda_3) \beta_1\beta_2 \beta_3 + \lambda_1 (1-\lambda_2) (1-\lambda_3) \\
& \beta_1\beta_2 \beta_3] f(r_1, r_1, s_1; s_2, s_2, s_2) + [(1-\lambda_1) \lambda_2\lambda_3 (1-\beta_1) (1-\beta_2) (1-\beta_3) + \lambda_1 (1-\lambda_2) \lambda_3 (1-\beta_1) \\
& (1-\beta_2) (1-\beta_3) + \lambda_1\lambda_2 (1-\lambda_3) (1-\beta_1) (1-\beta_2) (1-\beta_3)] f(r_1, s_1, s_1; r_2, r_2, r_2) + [(1-\lambda_1) \lambda_2\lambda_3 (1-\beta_1) \\
& (1-\beta_2) \beta_3 + \lambda_1 (1-\lambda_2) \lambda_3 (1-\beta_1) (1-\beta_2) \beta_3 + \lambda_1\lambda_2 (1-\lambda_3) (1-\beta_1) (1-\beta_2) \beta_3 + (1-\lambda_1) \lambda_2\lambda_3 \\
& (1-\beta_1) \beta_2 (1-\beta_3) + \lambda_1 (1-\lambda_2) \lambda_3 (1-\beta_1) \beta_2 (1-\beta_3) + \lambda_1\lambda_2 (1-\lambda_3) (1-\beta_1) \beta_2 (1-\beta_3) + [(1-\lambda_1) \\
& \lambda_2\lambda_3 \beta_1 (1-\beta_2) (1-\beta_3) + \lambda_1 (1-\lambda_2) \lambda_3 \beta_1 (1-\beta_2) (1-\beta_3) + \lambda_1\lambda_2 (1-\lambda_3) \beta_1 (1-\beta_2) (1-\beta_3)] f(r_1, \\
& s_1, s_1; s_2, r_2, r_2) + [(1-\lambda_1) \lambda_2\lambda_3 (1-\beta_1) \beta_2 \beta_3 + \lambda_1 (1-\lambda_2) \lambda_3 (1-\beta_1)\beta_2 \beta_3 + \lambda_1\lambda_2 (1-\lambda_3) (1-\beta_1) \\
& \beta_2 \beta_3 + (1-\lambda_1) \lambda_2\lambda_3 \beta_1 (1-\beta_2) \beta_3 + \lambda_1 (1-\lambda_2) \lambda_3 \beta_1 (1-\beta_2) \beta_3 + \lambda_1\lambda_2 (1-\lambda_3) \beta_1 (1-\beta_2) \beta_3 + (1-\lambda_1) \lambda_2\lambda_3 \\
& \beta_1 \beta_2 (1-\beta_3) + \lambda_1 (1-\lambda_2) \lambda_3 \beta_1\beta_2 (1-\beta_3) + \lambda_1\lambda_2 (1-\lambda_3) \beta_1 \beta_2 (1-\beta_3)] f(r_1, s_1, \\
& s_1; r_2, s_2, s_2) + [(1-\lambda_1) \lambda_2\lambda_3 \beta_1 \beta_2 \beta_3 + \lambda_1 (1-\lambda_2) \lambda_3 \beta_1\beta_2 \beta_3 + \lambda_1\lambda_2 (1-\lambda_3) \beta_1\beta_2 \beta_3] f(r_1, s_1, s_1
\end{aligned}$$

$$;s_2, s_2, s_2) + [\lambda_1 \lambda_2 \lambda_3 (1-\beta_1)(1-\beta_2)(1-\beta_3)] f(s_1, s_1, s_1; r_2, r_2, r_2) + [\lambda_1 \lambda_2 \lambda_3 (1-\beta_1)(1-\beta_2) \beta_3 + \lambda_1 \lambda_2 \lambda_3 (1-\beta_1) \beta_2 (1-\beta_3) + \lambda_1 \lambda_2 \lambda_3 \beta_1 (1-\beta_2)(1-\beta_3)] f(s_1, s_1, s_1; r_2, r_2, s_2) + [\lambda_1 \lambda_2 \lambda_3 \beta_1 (1-\beta_2) \beta_3 + \lambda_1 \lambda_2 \lambda_3 (1-\beta_1) \beta_2 \beta_3 + \lambda_1 \lambda_2 \lambda_3 \beta_1 \beta_2 (1-\beta_3)] f(s_1, s_1, s_1; r_2, s_2, s_2) + [\lambda_1 \lambda_2 \lambda_3 \beta_1 \beta_2 \beta_3] f(s_1, s_1, s_1; s_2, s_2, s_2) \} \quad (34)$$

The de-Casteljau algorithm is used to define the bi-cubic bipolynomial surface

$F_1(u, v)$ at each value of u from r_1 to s_1 , and v from r_2 to s_2 , where r_1, s_1, r_2 and s_2 belong to A .

Suppose $f_1(u_1, u_2, u_3; v_1, v_2, v_3) = F_1(u, v)$ and $[f(r_1, r_1, r_1; r_2, r_2, r_2), f(r_1, r_1, r_1; r_2, r_2, s_2), f(r_1, r_1, r_1; r_2, s_2, s_2), f(r_1, r_1, s_1; r_2, r_2, r_2), f(r_1, r_1, s_1; r_2, r_2, s_2), f(r_1, r_1, s_1; r_2, s_2, s_2), f(r_1, s_1, s_1; r_2, r_2, r_2), f(r_1, s_1, s_1; r_2, r_2, s_2), f(r_1, s_1, s_1; s_2, s_2, s_2), f(s_1, s_1, s_1; r_2, r_2, r_2), f(s_1, s_1, s_1; r_2, r_2, s_2), f(s_1, s_1, s_1; r_2, s_2, s_2), f(s_1, s_1, s_1; s_2, s_2, s_2)] = [p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}, p_{15}, p_{16}]$, are control points and equation (34) becomes :

$$F_1(u, v) = [(1-\lambda_1)(1-\lambda_2)(1-\lambda_3)(1-\beta_1)(1-\beta_2)(1-\beta_3)] p_1 + [(1-\lambda_1)(1-\lambda_2)(1-\lambda_3)(1-\beta_1)(1-\beta_2) \beta_3 + (1-\lambda_1)(1-\lambda_2)(1-\lambda_3)(1-\beta_1) \beta_2 (1-\beta_3) + (1-\lambda_1)(1-\lambda_2)(1-\lambda_3) \beta_1 (1-\beta_2)(1-\beta_3)] p_2 + [(1-\lambda_1)(1-\lambda_2)(1-\lambda_3)(1-\beta_1) \beta_2 \beta_3 + (1-\lambda_1)(1-\lambda_2)(1-\lambda_3) \beta_1 (1-\beta_2) \beta_3 + (1-\lambda_1)(1-\lambda_2)(1-\lambda_3) \beta_1 \beta_2 (1-\beta_3)] p_3 + [(1-\lambda_1)(1-\lambda_2)(1-\lambda_3) \beta_1 \beta_2 \beta_3] p_4 + [(1-\lambda_1)(1-\lambda_2) \lambda_3 (1-\beta_1)(1-\beta_2)(1-\beta_3) + (1-\lambda_1) \lambda_2 (1-\lambda_3)(1-\beta_1)(1-\beta_2)(1-\beta_3) + \lambda_1 (1-\lambda_2)(1-\lambda_3)(1-\beta_1)(1-\beta_2)(1-\beta_3)] p_5 + [(1-\lambda_1)(1-\lambda_2) \lambda_3 (1-\beta_1)(1-\beta_2) \beta_3 + (1-\lambda_1) \lambda_2 (1-\lambda_3)(1-\beta_1)(1-\beta_2) \beta_3 + \lambda_1 (1-\lambda_2)(1-\lambda_3)(1-\beta_1)(1-\beta_2) \beta_3 + (1-\lambda_1) \lambda_2 (1-\lambda_3)(1-\beta_1) \beta_2 (1-\beta_3) + (1-\lambda_1) \lambda_2 (1-\lambda_3)(1-\beta_1) \beta_2 (1-\beta_3) + \lambda_1 (1-\lambda_2)(1-\lambda_3) \beta_1 (1-\beta_2)(1-\beta_3)] p_6 + [(1-\lambda_1)(1-\lambda_2) \lambda_3 (1-\beta_1) \beta_2 \beta_3 + (1-\lambda_1) \lambda_2 (1-\lambda_3)(1-\beta_1) \beta_2 \beta_3 + \lambda_1 (1-\lambda_2)(1-\lambda_3) \beta_1 (1-\beta_2) \beta_3 + (1-\lambda_1) \lambda_2 (1-\lambda_3) \beta_1 \beta_2 (1-\beta_3) + (1-\lambda_1) \lambda_2 (1-\lambda_3) \beta_1 \beta_2 (1-\beta_3) + \lambda_1 (1-\lambda_2)(1-\lambda_3) \beta_1 \beta_2 (1-\beta_3)] p_7 + [(1-\lambda_1)(1-\lambda_2) \lambda_3 \beta_1 \beta_2 \beta_3 + (1-\lambda_1) \lambda_2 (1-\lambda_3) \beta_1 \beta_2 \beta_3 + \lambda_1 (1-\lambda_2)(1-\lambda_3) \beta_1 \beta_2 \beta_3] p_8 + [(1-\lambda_1) \lambda_2 \lambda_3 (1-\beta_1)(1-\beta_2)(1-\beta_3) + \lambda_1 (1-\lambda_2) \lambda_3 (1-\beta_1)(1-\beta_2)(1-\beta_3) + \lambda_1 \lambda_2 (1-\lambda_3)(1-\beta_1)(1-\beta_2)(1-\beta_3)] p_9 + [(1-\lambda_1) \lambda_2 \lambda_3 (1-\beta_1)(1-\beta_2) \beta_3 + \lambda_1 (1-\lambda_2) \lambda_3 (1-\beta_1)(1-\beta_2) \beta_3 + \lambda_1 \lambda_2 (1-\lambda_3)(1-\beta_1)(1-\beta_2) \beta_3 + (1-\lambda_1) \lambda_2 \lambda_3 (1-\beta_1) \beta_2 (1-\beta_3) + \lambda_1 (1-\lambda_2) \lambda_3 (1-\beta_1) \beta_2 (1-\beta_3) + \lambda_1 \lambda_2 (1-\lambda_3)(1-\beta_1) \beta_2 (1-\beta_3)] p_{10} + [(1-\lambda_1) \lambda_2 \lambda_3 (1-\beta_1) \beta_2 \beta_3 + \lambda_1 (1-\lambda_2) \lambda_3 (1-\beta_1) \beta_2 \beta_3 + \lambda_1 \lambda_2 (1-\lambda_3)(1-\beta_1) \beta_2 \beta_3 + (1-\lambda_1) \lambda_2 \lambda_3 \beta_1 (1-\beta_2) \beta_3 + \lambda_1 (1-\lambda_2) \lambda_3 \beta_1 (1-\beta_2) \beta_3 + \lambda_1 \lambda_2 (1-\lambda_3) \beta_1 (1-\beta_2) \beta_3 + (1-\lambda_1) \lambda_2 \lambda_3 \beta_1 \beta_2 (1-\beta_3) + \lambda_1 (1-\lambda_2) \lambda_3 \beta_1 \beta_2 (1-\beta_3) + \lambda_1 \lambda_2 (1-\lambda_3) \beta_1 \beta_2 (1-\beta_3)] p_{11} + [(1-\lambda_1) \lambda_2 \lambda_3 \beta_1 \beta_2 \beta_3 + \lambda_1 (1-\lambda_2) \lambda_3 \beta_1 \beta_2 \beta_3 + \lambda_1 \lambda_2 (1-\lambda_3) \beta_1 \beta_2 \beta_3] p_{12} + [\lambda_1 \lambda_2 \lambda_3 (1-\beta_1)(1-\beta_2)(1-\beta_3)] p_{13} + [\lambda_1 \lambda_2 \lambda_3 (1-\beta_1)(1-\beta_2) \beta_3 + \lambda_1 \lambda_2 \lambda_3 (1-\beta_1) \beta_2 (1-\beta_3) + \lambda_1 \lambda_2 \lambda_3 \beta_1 (1-\beta_2)(1-\beta_3)] p_{14} + [\lambda_1 \lambda_2 \lambda_3 \beta_1 (1-\beta_2) \beta_3 + \lambda_1 \lambda_2 \lambda_3 (1-\beta_1) \beta_2 \beta_3 + \lambda_1 \lambda_2 \lambda_3 \beta_1 \beta_2 (1-\beta_3)] p_{15} + [\lambda_1 \lambda_2 \lambda_3 \beta_1 \beta_2 \beta_3] p_{16}. \quad (35)$$

Eq (35) is called Gallier modified bi-cubic Bezier surfaces (3D), it is easily seen the above equation that the same properties if substitution of

$$\lambda_i = \frac{u_i - r_1}{s_1 - r_1}, 1 - \lambda_i = \frac{s_1 - u_i}{s_1 - r_1} \text{ and } \beta_j = \frac{v_j - r_2}{s_2 - r_2}, 1 - \beta_j = \frac{s_2 - v_j}{s_2 - r_2}, \text{ in equations (35)}$$

where $r_1 < s_1$, and $r_2 < s_2$ for u_i and $v_j \in R$, (where R is real number), r_1, s_1, r_2 and $s_2 \in A$, $r_1 \leq u_i \leq s_1$ and $r_2 \leq v_j \leq s_2$, (for $i, j = 1, 2, 3$).

Note II

The coefficients of the control points in the be-cubic Bezier surface (35) are called Bernstein Polynomials given as:

$$\{ [(1-\lambda_1)(1-\lambda_2)(1-\lambda_3)(1-\beta_1)(1-\beta_2)(1-\beta_3)] + [(1-\lambda_1)(1-\lambda_2)(1-\lambda_3)(1-\beta_1)(1-\beta_2) \beta_3 + (1-\lambda_1)(1-\lambda_2)(1-\lambda_3)(1-\beta_1) \beta_2 (1-\beta_3) + (1-\lambda_1)(1-\lambda_2)(1-\lambda_3) \beta_1 (1-\beta_2)(1-\beta_3)] + [(1-\lambda_1)(1-\lambda_2)(1-\lambda_3)(1-\beta_1) \beta_2 \beta_3 + (1-\lambda_1)(1-\lambda_2)(1-\lambda_3) \beta_1 (1-\beta_2) \beta_3 + (1-\lambda_1)(1-\lambda_2)(1-\lambda_3) \beta_1 \beta_2 (1-\beta_3)] + [(1-\lambda_1)(1-\lambda_2)(1-\lambda_3) \beta_1 \beta_2 \beta_3] + [(1-\lambda_1)(1-\lambda_2) \lambda_3 (1-\beta_1)(1-\beta_2)(1-\beta_3) + (1-\lambda_1) \lambda_2 (1-\lambda_3)(1-\beta_1)(1-\beta_2)(1-\beta_3) + \lambda_1 (1-\lambda_2)(1-\lambda_3)(1-\beta_1)(1-\beta_2)(1-\beta_3)] \}$$

$$\begin{aligned}
 & \beta_1)(1-\beta_2)(1-\beta_3) + \lambda_1(1-\lambda_2)(1-\lambda_3)(1-\beta_1)(1-\beta_2)(1-\beta_3)] + [(1-\lambda_1)(1-\lambda_2)\lambda_3(1-\beta_1)(1-\beta_2) \\
 & \beta_3 + (1-\lambda_1)\lambda_2(1-\lambda_3)(1-\beta_1)(1-\beta_2)\beta_3 + \lambda_1(1-\lambda_2)(1-\lambda_3)(1-\beta_1)(1-\beta_2)\beta_3 + (1-\lambda_1)(1-\lambda_2)\lambda_3(1- \\
 & \beta_1)\beta_2(1-\beta_3) + (1-\lambda_1)\lambda_2(1-\lambda_3)(1-\beta_1)\beta_2(1-\beta_3) + \lambda_1(1-\lambda_2)(1-\lambda_3)(1-\beta_1)\beta_2(1-\beta_3) + (1-\lambda_1)(1- \\
 & \lambda_2)\lambda_3\beta_1(1-\beta_2)(1-\beta_3) + (1-\lambda_1)\lambda_2(1-\lambda_3)\beta_1(1-\beta_2)(1-\beta_3) + \lambda_1(1-\lambda_2)(1-\lambda_3)\beta_1(1-\beta_2)(1-\beta_3)] \\
 & + [(1-\lambda_1)(1-\lambda_2)\lambda_3(1-\beta_1)\beta_2\beta_3 + (1-\lambda_1)\lambda_2(1-\lambda_3)(1-\beta_1)\beta_2\beta_3 + \lambda_1(1-\lambda_2)(1-\lambda_3)(1-\beta_1)\beta_2\beta_3 \\
 & + (1-\lambda_1)(1-\lambda_2)\lambda_3\beta_1(1-\beta_2)\beta_3 + (1-\lambda_1)\lambda_2(1-\lambda_3)\beta_1(1-\beta_2)\beta_3 + \lambda_1(1-\lambda_2)(1-\lambda_3)\beta_1(1-\beta_2)\beta_3 \\
 & + (1-\lambda_1)(1-\lambda_2)\lambda_3\beta_1\beta_2(1-\beta_3) + (1-\lambda_1)\lambda_2(1-\lambda_3)\beta_1\beta_2(1-\beta_3) + \lambda_1(1-\lambda_2)(1-\lambda_3)\beta_1\beta_2(1-\beta_3)] \\
 & + [(1-\lambda_1)(1-\lambda_2)\lambda_3\beta_1\beta_2\beta_3 + (1-\lambda_1)\lambda_2(1-\lambda_3)\beta_1\beta_2\beta_3 + \lambda_1(1-\lambda_2)(1-\lambda_3)\beta_1\beta_2\beta_3] + [(1-\lambda_1)\lambda_2\lambda_3 \\
 & (1-\beta_1)(1-\beta_2)(1-\beta_3) + \lambda_1(1-\lambda_2)\lambda_3(1-\beta_1)(1-\beta_2)(1-\beta_3) + \lambda_1\lambda_2(1-\lambda_3)(1-\beta_1)(1-\beta_2)(1- \\
 & \beta_3)] + [(1-\lambda_1)\lambda_2\lambda_3(1-\beta_1)(1-\beta_2)\beta_3 + \lambda_1(1-\lambda_2)\lambda_3(1-\beta_1)(1-\beta_2)\beta_3 + \lambda_1\lambda_2(1-\lambda_3)(1-\beta_1)(1- \\
 & \beta_2)\beta_3 + (1-\lambda_1)\lambda_2\lambda_3(1-\beta_1)\beta_2(1-\beta_3) + \lambda_1(1-\lambda_2)\lambda_3(1-\beta_1)\beta_2(1-\beta_3) + \lambda_1\lambda_2(1-\lambda_3)(1-\beta_1) \\
 & \beta_2(1-\beta_3) + [(1-\lambda_1)\lambda_2\lambda_3\beta_1(1-\beta_2)(1-\beta_3) + \lambda_1(1-\lambda_2)\lambda_3\beta_1(1-\beta_2)(1-\beta_3) + \lambda_1\lambda_2(1-\lambda_3)\beta_1 \\
 & (1-\beta_2)(1-\beta_3)] + [(1-\lambda_1)\lambda_2\lambda_3(1-\beta_1)\beta_2\beta_3 + \lambda_1(1-\lambda_2)\lambda_3(1-\beta_1)\beta_2\beta_3 + \lambda_1\lambda_2(1-\lambda_3)(1-\beta_1)\beta_2 \\
 & \beta_3 + (1-\lambda_1)\lambda_2\lambda_3\beta_1(1-\beta_2)\beta_3 + \lambda_1(1-\lambda_2)\lambda_3\beta_1(1-\beta_2)\beta_3 + \lambda_1\lambda_2(1-\lambda_3)\beta_1(1-\beta_2)\beta_3 + (1-\lambda_1) \\
 & \lambda_2\lambda_3\beta_1\beta_2(1-\beta_3) + \lambda_1(1-\lambda_2)\lambda_3\beta_1\beta_2(1-\beta_3) + \lambda_1\lambda_2(1-\lambda_3)\beta_1\beta_2(1-\beta_3)] + [(1-\lambda_1)\lambda_2\lambda_3\beta_1\beta_2 \\
 & \beta_3 + \lambda_1(1-\lambda_2)\lambda_3\beta_1\beta_2\beta_3 + \lambda_1\lambda_2(1-\lambda_3)\beta_1\beta_2\beta_3] + [\lambda_1\lambda_2\lambda_3(1-\beta_1)(1-\beta_2)(1-\beta_3)] + [\lambda_1\lambda_2\lambda_3(1- \\
 & \beta_1)(1-\beta_2)\beta_3 + \lambda_1\lambda_2\lambda_3(1-\beta_1)\beta_2(1-\beta_3) + \lambda_1\lambda_2\lambda_3\beta_1(1-\beta_2)(1-\beta_3)] + [\lambda_1\lambda_2\lambda_3\beta_1(1-\beta_2) \\
 & \beta_3 + \lambda_1\lambda_2\lambda_3(1-\beta_1)\beta_2\beta_3 + \lambda_1\lambda_2\lambda_3\beta_1\beta_2(1-\beta_3)] + [\lambda_1\lambda_2\lambda_3\beta_1\beta_2\beta_3] \} = \{ [(1-\lambda_1) + \lambda_1][(1- \\
 & \lambda_2) + \lambda_2][(1-\lambda_3) + \lambda_3][(1-\beta_1) + \beta_1][(1-\beta_2) + \beta_2][(1-\beta_3) + \beta_3] \} = \left\{ \left[\left(\frac{s_1 - u_1}{s_1 - r_1} \right) + \right. \right. \\
 & \left. \left. \frac{u_1 - r_1}{s_1 - r_1} \right] \left[\left(\frac{s_1 - u_2}{s_1 - r_1} \right) + \frac{u_2 - r_1}{s_1 - r_1} \right] \left[\left(\frac{s_1 - u_3}{s_1 - r_1} \right) + \frac{u_3 - r_1}{s_1 - r_1} \right] \left[\left(\frac{s_2 - v_1}{s_2 - r_2} \right) + \frac{v_1 - r_2}{s_2 - r_2} \right] \right. \\
 & \left. \left(\frac{s_2 - v_2}{s_2 - r_2} \right) + \frac{v_2 - r_2}{s_2 - r_2} \right] \left[\left(\frac{s_2 - v_3}{s_2 - r_2} \right) + \frac{v_3 - r_2}{s_2 - r_2} \right] \right\} = 1, \quad (36)
 \end{aligned}$$

which is called Bernstein bipolynomial bi-cubic (3D)) [Gallier 00]. The terms are obtained by expanding the expression and then collecting terms in the various product. This equation immediately yields an important property of these polynomials. They are added to unity at every u and v .

What's good about the algorithm is that it does not assume any prior knowledge of the curve. All that is given is the sequence of control points $P_1 \dots P_{16}$, of $m+1$ to define 3D surfaces (Mathematically they are said to be generated from the Cartesian product of two curves). A bi-cubic Bezier surfaces. The essence of the de-Casteljau algorithm is to compute $f(u_1, u_2, u_3; v_1, v_2, v_3)$, by repeated linear interpolations using the fact that f is symmetric and affine.

These control points play a major role in the de-Casteljau algorithm and its extensions. The bipolynomial design defining F passes through the two points p_0 and p_{16} , but not through the other control points. If $u_1 = u_2 = u_3 = u$, and $v_1 = v_2 = v_3 = v$ gives $\lambda = \lambda_1 = \lambda_2 = \lambda_3$, and $\beta = \beta_1 = \beta_2 = \beta_3$ Eq (35) becomes: -

$$\begin{aligned}
 F_1(u, v) = & [(1-\lambda)^3(1-\beta)^3]p_1 + 3[(1-\lambda)^3(1-\beta)^2\beta]p_2 + 3[(1-\lambda)^3(1-\beta)\beta^2]p_3 + [(1-\lambda)^3\beta^3]p_4 \\
 & + 3[(1-\lambda)^2\lambda(1-\beta)^3]p_5 + 9[(1-\lambda)^2\lambda(1-\beta)^2\beta]p_6 + 9[(1-\lambda)^2\lambda(1-\beta)\beta^2]p_7 + \\
 & 3[(1-\lambda)^2\lambda\beta^3]p_8 + 3[(1-\lambda)\lambda^2(1-\beta)^3]p_9 + 9[(1-\lambda)\lambda^2(1-\beta)^2\beta]p_{10} + 9[(1-\lambda)\lambda^2(1-\beta) \\
 & \beta^2]p_{11} + 3[(1-\lambda)\lambda^2\beta^3]p_{12} + [\lambda^3(1-\beta)^3]p_{13} + 3[\lambda^3(1-\beta)^2\beta]p_{14} + 3[\lambda^3(1-\beta)\beta^2]p_{15} \\
 & + [\lambda^3\beta^3]p_{16}
 \end{aligned}$$

Or

$$\begin{aligned}
 F_1(u, v) = & \sum_{i=0}^3 \sum_{j=0}^3 \binom{3}{i} \binom{3}{j} [1-\lambda]^{3-i} [\lambda]^i [1-\beta]^{3-j} [\beta]^j P_{ij} \\
 \text{Or} & \sum_{i=0}^3 \sum_{j=0}^3 \binom{3}{i} \binom{3}{j} \left[\frac{s_1 - u}{s_1 - r_1} \right]^{3-i} \left[\frac{u - r_1}{s_1 - r_1} \right]^i \left[\frac{s_2 - v}{s_2 - r_2} \right]^{3-j} \left[\frac{v - r_2}{s_2 - r_2} \right]^j P_{ij} \quad (37)
 \end{aligned}$$

Where

$$\binom{3}{i} = \frac{3!}{i!(3-i)!}, \quad \binom{3}{j} = \frac{3!}{j!(3-j)!}$$

. For $r_1=0, s_1=1$, and $r_2=0, s_2=1 \rightarrow \lambda = u$, and $\beta = v$ in Eq (37) become: -

$$F_1(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 \binom{3}{i} \binom{3}{j} [1-u]^{3-i} [u]^i [1-v]^{3-j} [v]^j P_{ij} \quad (38)$$

Eq (38) called original cubic Bezier surfaces is dependent on frame $[0, 1]^2$. It is identical to Eq (15). [where $(p_{00}, p_{01}, p_{02}, p_{03}, p_{10}, p_{11}, p_{12}, p_{13}, p_{20}, p_{21}, p_{22}, p_{23}, p_{30}, p_{31}, p_{32}, p_{33}) = (P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9, P_{10}, P_{11}, P_{12}, P_{13}, P_{14}, P_{15}, P_{16})$.

To explain the above a bipolynomial new surface, (3D) of degree $\langle 3, 3 \rangle$, the following example is given in a set of control points

$p_i = (x_i, y_i, z_i)$ for $i=1, 2, \dots, 16$.

Treat the coordinates of each point as a three-component vector. That is

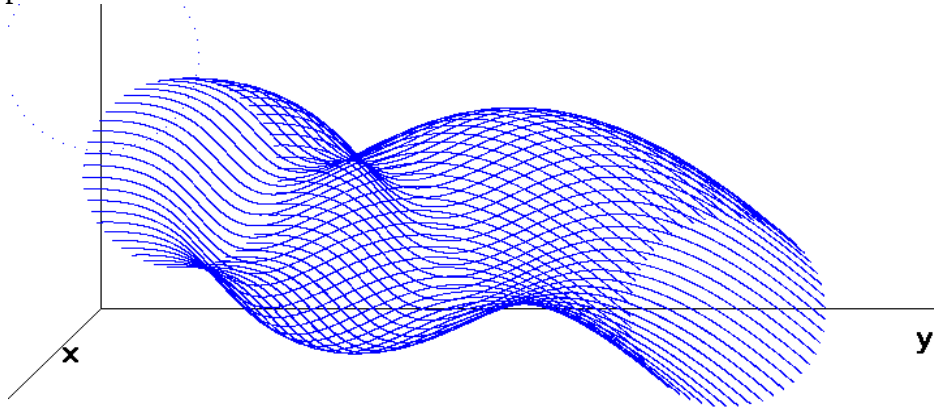
$$P_i = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}.$$

The set of points, in parametric form is

$$F(u, v) = \begin{pmatrix} F_1(u, v) \\ F_2(u, v) \\ F_3(u, v) \end{pmatrix} = \begin{pmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \end{pmatrix} \quad r_1 \leq u \leq s_1 \text{ and } r_2 \leq v \leq s_2$$

First-Technique

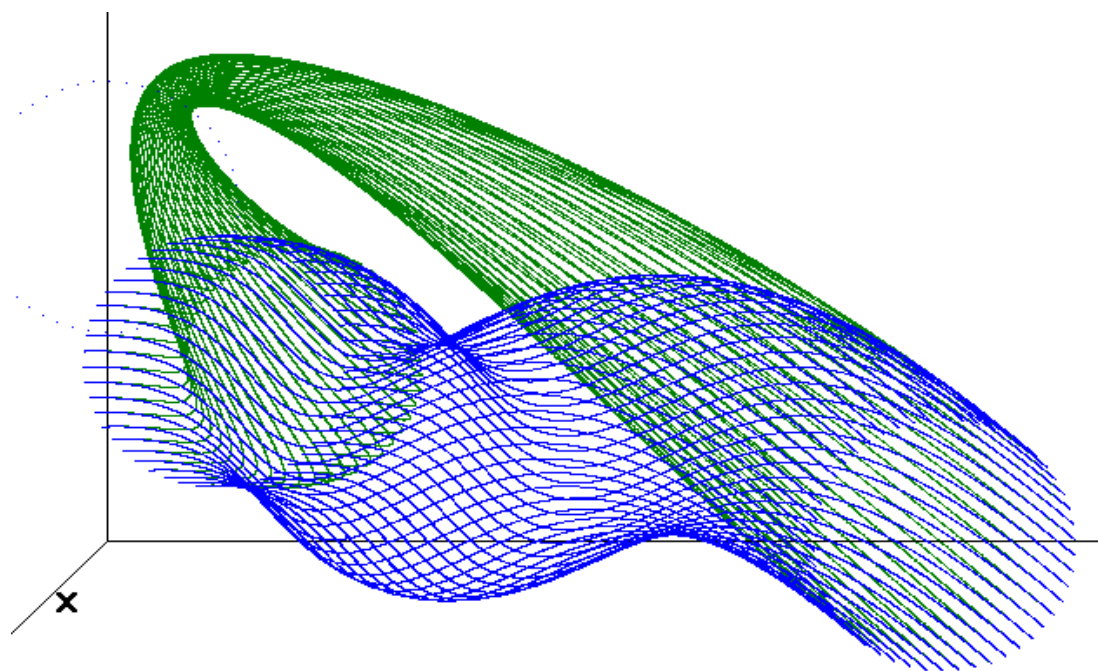
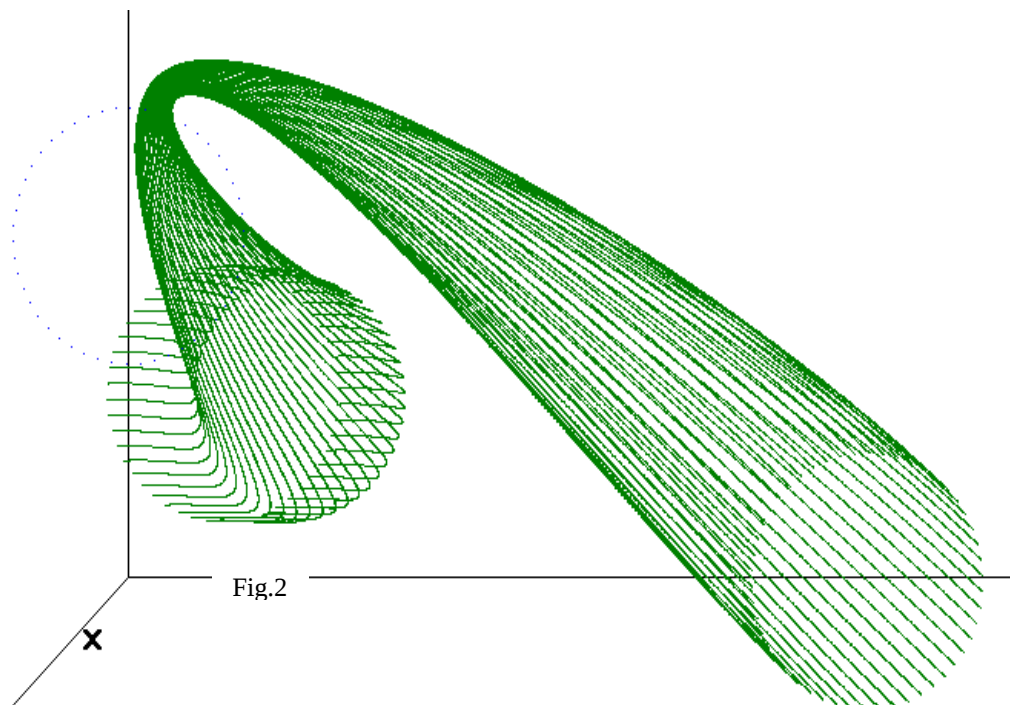
In this mathematical model suppose $\lambda_1=\lambda_2=\lambda_3=\lambda$ and $\beta_3=\beta_2=\beta_1=\beta$, then Eq (35) reduces to the original Bezier surface in Eq (15) see fig1. Eq (35) is built by mathematically developing de-Casteljau algorithm. This can be seen by simply mathematically changing Eq (35) which reduces to modified Eq (37), which easily reduces to original Eq (38) by taking special case for $r_1=0, s_1=1$, and $r_2=0, s_2=1$. Thus shown the base classical populates of Bezier surface are interpolations at first point P_1 and last point P_{16} . The design **changes** only when **change at least one or more** of control points of Bezier surface.



. Fig.1. Bezier surface which gives the base classical populates of Bezier surface is interpolations at first point P_1 and last point P_{16} . The design **changes** only when **change at least one or more** of control points of Bezier surface.

Second-Technique

In this mathematical model suppose u_1 or u_2 or u_3 , is taken to increase, gives λ_1 or λ_2 or λ_3 be increase in Eq (35). It is found in this case that the design can be moved to **above (left)**, with respect to value of u_1 or u_2 or u_3 , with **no need to change** any of the control points, see figs 2, and 3.

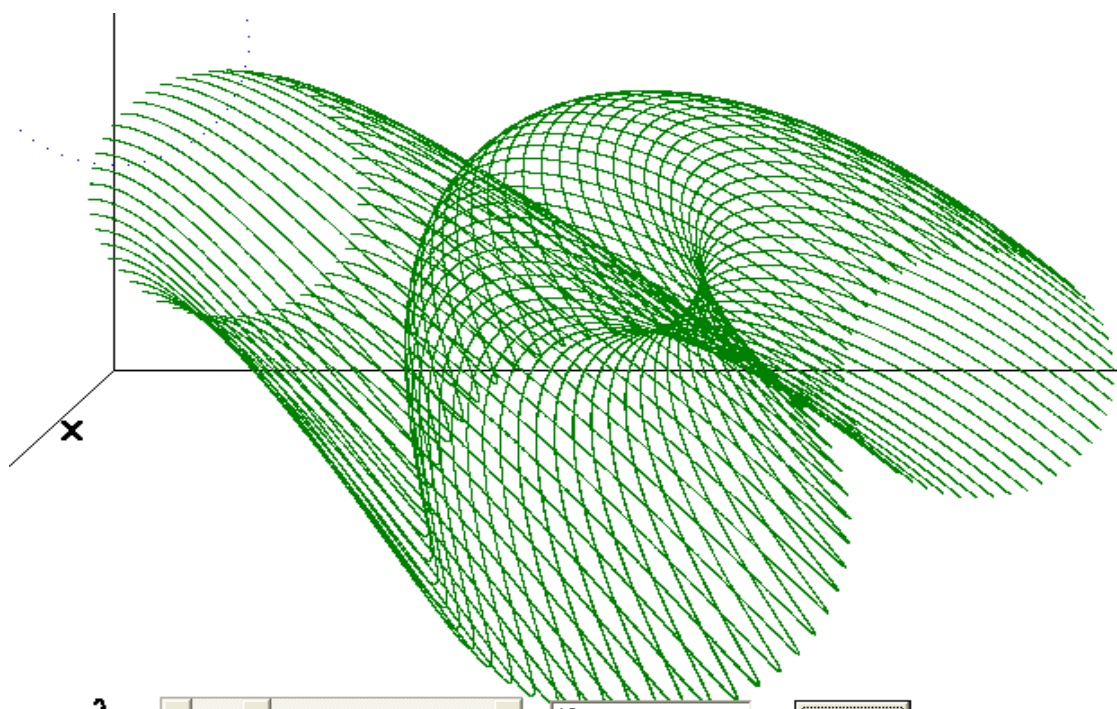


. Figs.2,3. Suppose λ_1 or λ_2 or λ_3 , in Bezier surface when u_1 or u_2 or u_3 , increases the design can be moved to **above (left)**, with no need to change any of the control points.

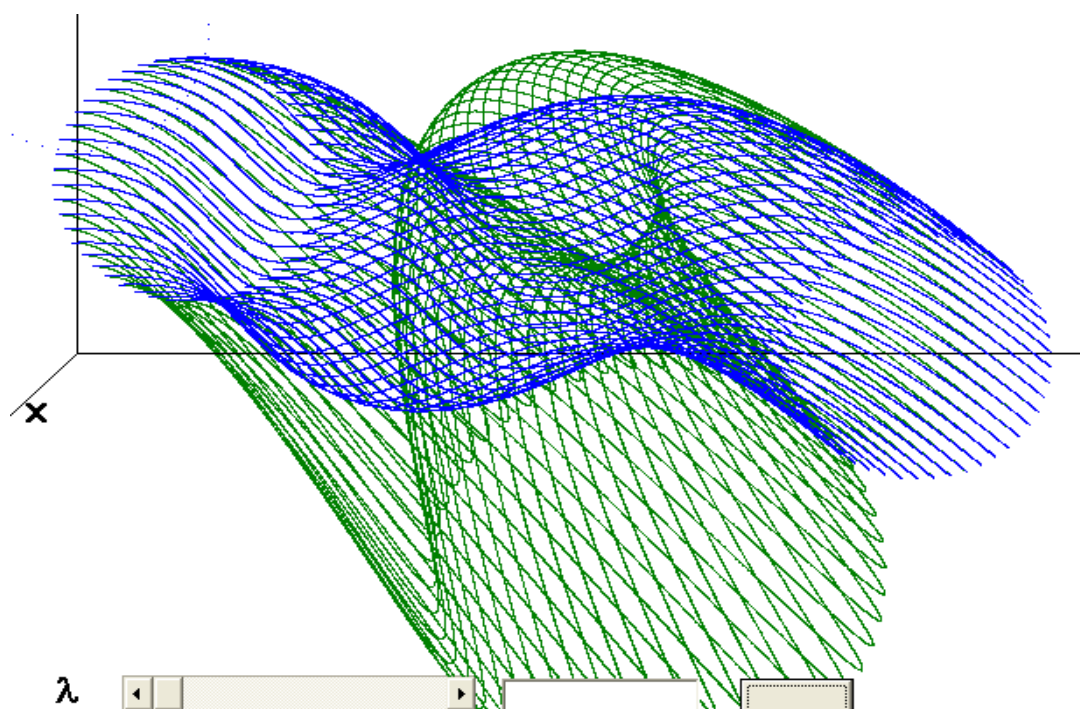
Third -Technique

In this mathematical model suppose v_1 or v_2 or v_3 is taken to increase, gives β_1 or β_2 or β_3 , be increase, it is found in this case that the design can be moved to

(below)dowe (right) with respecte to the value of v_1 or v_2 or v_3 with **no need to change** any of the control points, see figs 4,and5.



. Fig.4



. Figs.4,5. . Suppose β_1 or β_2 or β_3 , in Bezier surface when v_1 or v_2 or v_3 increases the design can be moved **(below)down (right)** with no need to change any of the control points.

Fourth -Technique

In this mathematical model suppose u_1 and v_1 is taken to increase, λ_1 , and β_1 , gives the design can be moved to **above (left)**, and **(below)down (right)** with **no need to change** any of the control points. See fig 4. Another property, which is obtained from the new mathematical equation of Bezier, surface (35) at change it to the new condition as this case. Note that the building of Bezier bicubic in Eq (26) is given by Gallier modified de Casteljau algorithm by successive mathematical steps until the modified formula (35), is given which is the difference from initial Eq (15) and Gallier modified Eq (37). Fig 6.

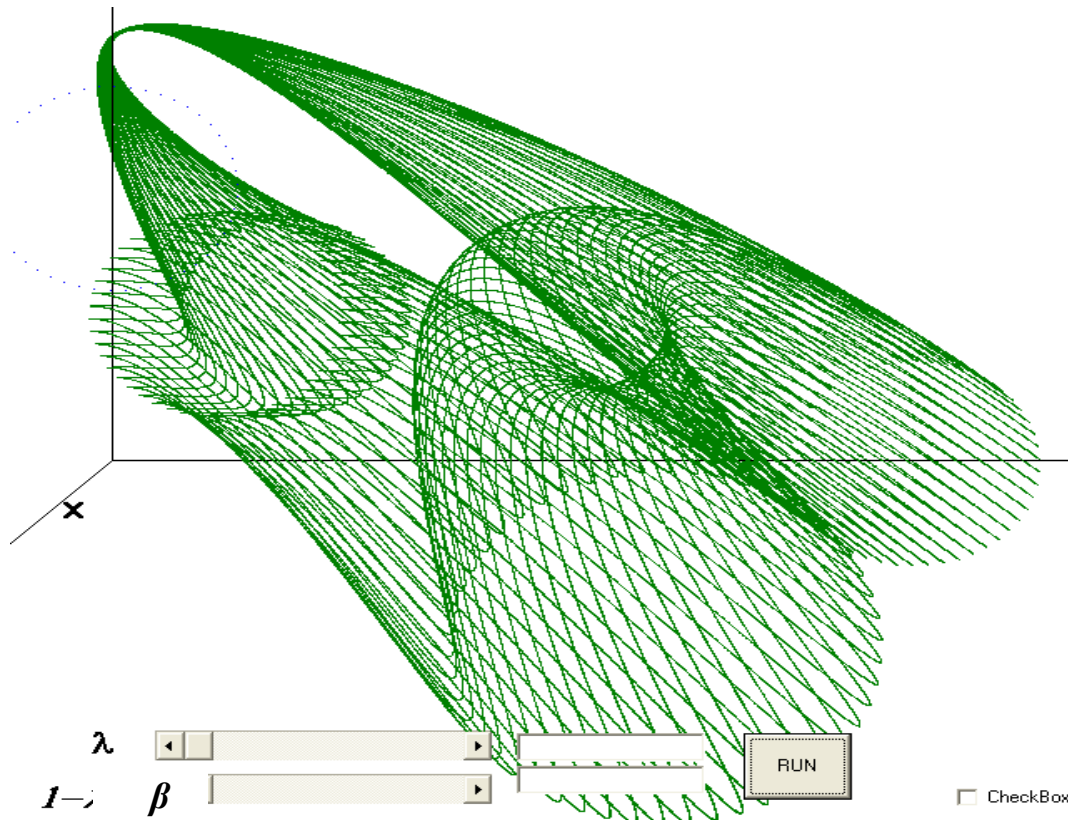


Fig.6. suppose λ_1 , and β_1 , in Bezier surface when u_1 and v_1 increase the design can be moved **above (left)**, and **(below)down (right)** with no need to change any of the control points.

Discussion

In this work the study of a bipolynomial surface, (3D) of degree $\langle 3, 3 \rangle$, is based on:-

- In computer graphics one can generally use degree three. Quadratic curves are not flexible enough and going above degree 3 gives rise to complications and so the choice of cubic is the best compromise for most computer graphics applications.
- Defining a bipolynomial surface in terms, of polar forms. A bipolynomial surface involved contains two variables, the natural way to polarize polynomial surface. The approach yields bipolynomial surface {also called tensor product surfaces}. It is shown how versions of the de-Casteljau algorithm can be turned into subdivisions.
- Showing how to compute a new control net from given net, depending on the parameters. This is one of indications that dealing with surface is far more complex than dealing with curves.
- The linear form of a modified Bezier formula of a bipolynomial surface, (3D). In Eq (35), which leads the designer to approach the equation, contains 64 net, form 16

control points. The algorithm which allows the designer to produce a design in combinational way allows him to get the shape that he has in his mind keeping the 16 control point for 3D design.

- The notion of Bezier curves to define Bezier surface. Typically, a Bezier surface parameterized by variables u and v , using the range over $[r_1, s_1]$, and $[r_2, s_2]$, is see Eq (35), instead of both ranges over the interval $[0,1]$. See Eqs (15 and 38), which are a special case of our modified Eq (35).
- The modification for Controlling, and generating the design, in this paper gives changing of u_i and v_j (for $i, j = 1, 2, 3$), in the values of the coefficient of parameters of any point. See figs 2, 3, 4, 5 and 6.
- The first and last control points are the end points of curve segment.
- The bipolynomial design defined F passes through the two points p_0 and p_{16} , but not through the other control points
- The modified linear mathematical construction of the equation gives the designer more room to control and construct his design. This is done through controlling certain parts of the equation which give eventually great effect on certain parts of the design or the design at large.

Conclusions

There are several advantages, of modified Bezier surfaces. These include the following:

- This work concludes a bipolynomial surface is based on a mathematical procedure depending on the linear construction of a bipolynomial surface and following de-Casteljau algorithm. The method shows a great flexibility in 3D design controlling area with no need to change the control points of the design.
- An important property from the point of view of the algorithm that deals with surfaces is that a curve is always enclosed in the convex hull formed by the control polygon of a 3D space curve and can be considered to be the polygon formed by placing an elastic band around the control points. This follows from the fact that the basic comes from the sum to unity for all u and v . See Eq (36).
- The designer has the advantage of controlling and modifying the design without changing any of the control points. This method can be seen more efficient in designs in comparison with that needed in conventional methods.

References.

- Buss Samuel R. 3-D (2003) Computer Graphics A Mathematical Introduction with Open GL.
- Faux I D (1983) Computational Geometry for Design and manufacture, Ellis Hood ltd.
- Gallier Jean (2000) Curves and Surfaces in Geometric Modeling Theory and Algorithms. Morgan Kaufmann publishers.
- Gerald C. F and Wheatley P. O. (1999) Applied Numerical Analysis. Addison Wesley.
- Hill F. S, Jr (2001) Computer Graphics Using Open GL. Prentice Hall.
- Jaber A. Mohssin (2010) A Proposed Algorithm to Improve The Time of Cubic Bezier Curve. First Computer Science conference Proceedings. University of Technology.
- Jaber A. Mohssin (2005) Modified Curves Mathematical Models. Ph D thesis. University of Technology.
- Watt Alan (2000) "3D Computer Graphics", Published by Addison Wesley Company, Inc.
- Klawonn Frank (2008) Introduction to Computer Graphics Using Java 2D and 3D Springer-Verlag London Limited.
- Lengyel Eric (2004) Mathematics for 3D Game Programming and Computer Graphics Charles River Media, Inc.