



Last updated: Yeh-Liang Hsu (2010-11-18).

Note: This is the course material for “ME550 Geometric modeling and computer graphics,” Yuan Ze University. Part of this material is adapted from *CAD/CAM Theory and Practice*, by Ibrahim Zeid, McGraw-Hill, 1991. This material is be used strictly for teaching and learning of this course.

Synthetic surfaces

1. Bezier surface

Two sets of orthogonal Bezier curves can be used to design an object surface by specifying by an input mesh of control points, as shown in Figure 1. Bezier surfaces have the same properties as Bezier curves, and they provide a convenient method for interactive design applications. For each surface patch, we can select a mesh of control points in the xy “ground” plane, then we choose elevations above the ground plane for the z -coordinate values of the control points. Patches can then be pieced together using the boundary constraints.

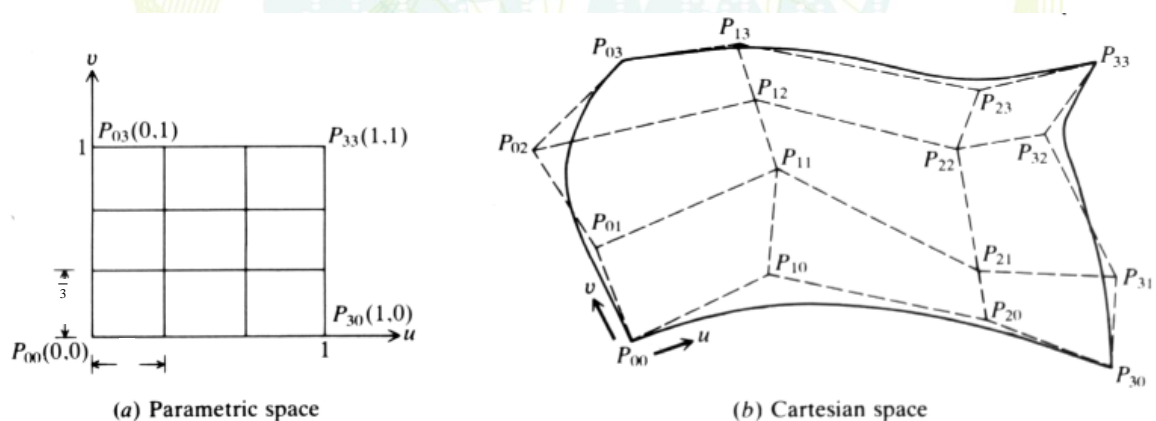


Figure 1. A cubic Bezier patch.

◇ Assignment 1

Generate the surface of an object, e.g., a car, the mouse of your computer, etc., using your CAD software. Use 4×4 control points as your input. Define an xy “ground plane” first, then use the procedure described above to adjust the z -coordinates of your control points, until you get the surface you want. Can you render this surface in your CAD software? ◇

A **tensor product Bezier surface** is an extension for the Bezier curve in two parametric directions u and v .

$$\mathbf{P}(u, v) = \sum_{i=0}^n \sum_{j=0}^m \mathbf{P}_{ij} B_{i,n}(u) B_{j,m}(v), \quad 0 \leq u \leq 1, \quad 0 \leq v \leq 1 \quad (1)$$

where $\mathbf{P}(u, v)$ is any point on the surface and \mathbf{P}_{ij} are the control points. These points form the vertices of the control or characteristic polyhedron.

Figure 1 shows a cubic Bezier patch. Substituting $n=3$ and $m=3$ into Equation (1), the patch equation is

$$\mathbf{P}(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 \mathbf{P}_{ij} B_{i,3}(u) B_{j,3}(v), \quad 0 \leq u \leq 1, \quad 0 \leq v \leq 1 \quad (2)$$

This equation can be expanded to give

$$\begin{aligned} \mathbf{P}(u, v) &= \sum_{i=0}^3 B_{i,3}(u) [\mathbf{P}_{i0} B_{0,3}(v) + \mathbf{P}_{i1} B_{1,3}(v) + \mathbf{P}_{i2} B_{2,3}(v) + \mathbf{P}_{i3} B_{3,3}(v)] \\ &= B_{0,3}(u) [\mathbf{P}_{00} B_{0,3}(v) + \mathbf{P}_{01} B_{1,3}(v) + \mathbf{P}_{02} B_{2,3}(v) + \mathbf{P}_{03} B_{3,3}(v)] \\ &\quad + B_{1,3}(u) [\mathbf{P}_{10} B_{0,3}(v) + \mathbf{P}_{11} B_{1,3}(v) + \mathbf{P}_{12} B_{2,3}(v) + \mathbf{P}_{13} B_{3,3}(v)] \\ &\quad + B_{2,3}(u) [\mathbf{P}_{20} B_{0,3}(v) + \mathbf{P}_{21} B_{1,3}(v) + \mathbf{P}_{22} B_{2,3}(v) + \mathbf{P}_{23} B_{3,3}(v)] \\ &\quad + B_{3,3}(u) [\mathbf{P}_{30} B_{0,3}(v) + \mathbf{P}_{31} B_{1,3}(v) + \mathbf{P}_{32} B_{2,3}(v) + \mathbf{P}_{33} B_{3,3}(v)] \end{aligned}$$

This equation can be written in a matrix form as

$$\mathbf{P}(u, v) = \begin{bmatrix} B_{0,3}(u) & B_{1,3}(u) & B_{2,3}(u) & B_{3,3}(u) \end{bmatrix} \begin{bmatrix} \mathbf{P}_{00} & \mathbf{P}_{01} & \mathbf{P}_{02} & \mathbf{P}_{03} \\ \mathbf{P}_{10} & \mathbf{P}_{11} & \mathbf{P}_{12} & \mathbf{P}_{13} \\ \mathbf{P}_{20} & \mathbf{P}_{21} & \mathbf{P}_{22} & \mathbf{P}_{23} \\ \mathbf{P}_{30} & \mathbf{P}_{31} & \mathbf{P}_{32} & \mathbf{P}_{33} \end{bmatrix} \begin{bmatrix} B_{0,3}(v) \\ B_{1,3}(v) \\ B_{2,3}(v) \\ B_{3,3}(v) \end{bmatrix} \quad (3)$$

or

$$\mathbf{P}(u, v) = \begin{bmatrix} (1-u)^3 & 3u(1-u)^2 & 3u^2(1-u) & u^3 \end{bmatrix} [\mathbf{P}] \begin{bmatrix} (1-v)^3 \\ 3v(1-v)^2 \\ 3v^2(1-v) \\ v^3 \end{bmatrix}$$

$$\mathbf{P}(u, v) = \mathbf{U}^T [\mathbf{M}_B] [\mathbf{P}] [\mathbf{M}_B]^T \mathbf{V} \quad (4)$$

◇ Assignment 2

Use the same 4×4 control points generated in assignment 1, write a Matlab program to draw the Bezier surface using Equation (4). Use proper mesh and views to display the surface. Compare the surface you draw with the surface generated in assignment 1. Show the Matlab program too. ◇

2. B-Spline surface

A tensor product B-spline surface is an extension for the B-spline curve in two parametric directions u and v .

$$\mathbf{P}(u, v) = \sum_{i=0}^n \sum_{j=0}^m \mathbf{P}_{ij} N_{i,k}(u) N_{j,l}(v), \quad 0 \leq u \leq u_{\max}, 0 \leq v \leq v_{\max} \quad (5)$$

For $n=m=3$, the equivalent bicubic formulation of an open and closed cubic B-spline surface can be derived as below. First, let us find the matrix form of Equation (5). This equation is identical in form to the Bezier surface equation (1). Thus, by replacing the Bernstein polynomials in Equation (4) by the B-spline functions yields the matrix form of Equation (5) as

$$\mathbf{P}(u, v) = \begin{bmatrix} N_{0,k}(u) & N_{1,k}(u) & \cdots & N_{n,k}(u) \end{bmatrix} \begin{bmatrix} \mathbf{P}_{00} & \mathbf{P}_{01} & \cdots & \mathbf{P}_{0m} \\ \mathbf{P}_{10} & \mathbf{P}_{11} & \cdots & \mathbf{P}_{1m} \\ \vdots & \vdots & & \vdots \\ \mathbf{P}_{n0} & \mathbf{P}_{n1} & \cdots & \mathbf{P}_{nm} \end{bmatrix} \begin{bmatrix} N_{0,l}(v) \\ N_{1,l}(v) \\ \vdots \\ N_{m,l}(v) \end{bmatrix} \quad (6)$$

or

$$\mathbf{P}(u, v) = \begin{bmatrix} N_{0,k}(u) & N_{1,k}(u) & \cdots & N_{n,k}(u) \end{bmatrix} [\mathbf{P}] \begin{bmatrix} N_{0,l}(v) \\ N_{1,l}(v) \\ \vdots \\ N_{m,l}(v) \end{bmatrix} \quad (7)$$

where $[\mathbf{P}]$ is an $(n+1) \times (m+1)$ matrix of the vertices of the characteristic polyhedron of the B-spline surface patch. For a 4×4 cubic B-spline patch, Equation (6) becomes

$$\mathbf{P}(u, v) = \begin{bmatrix} N_{0,4}(u) & N_{1,4}(u) & N_{2,4}(u) & N_{3,4}(u) \end{bmatrix} \begin{bmatrix} \mathbf{P}_{00} & \mathbf{P}_{01} & \mathbf{P}_{02} & \mathbf{P}_{03} \\ \mathbf{P}_{10} & \mathbf{P}_{11} & \mathbf{P}_{12} & \mathbf{P}_{13} \\ \mathbf{P}_{20} & \mathbf{P}_{21} & \mathbf{P}_{22} & \mathbf{P}_{23} \\ \mathbf{P}_{30} & \mathbf{P}_{31} & \mathbf{P}_{32} & \mathbf{P}_{33} \end{bmatrix} \begin{bmatrix} N_{0,4}(v) \\ N_{1,4}(v) \\ N_{2,4}(v) \\ N_{3,4}(v) \end{bmatrix}$$

◇ Assignment 3

Redo assignment 1. This time use as many control points as necessary to generate a better-looking surface. ◇

3. Coons Surface

All the surface methods introduced thus far share one common philosophy; that is, they all require a finite number of data points to generate the respective surfaces. In contrast, **a Coons surface patch is a form of “transfinite interpolation,” which indicates that the Coons scheme interpolates to an infinite number of data points,** that is, to all points of a curve segment, to generate the surface. The Coons patch is particularly useful in blending four prescribed intersecting curves, which form a closed boundary.

Development of the Coons surface patch centers around answering the following question: what is a suitable well-behaved function $\mathbf{P}(u, v)$ which blend the four given boundary curves and which satisfies the boundary conditions, that is reduces to the correct boundary curve when $u=0$, $u=1$, $v=0$, and $v=1$.

Let us first consider the case of a **bilinearly blended Coons patch**

$$\mathbf{P}_1(u, v) = (1-u)\mathbf{P}(0, v) + u\mathbf{P}(1, v) \quad (8)$$

$$\mathbf{P}_2(u, v) = (1-v)\mathbf{P}(u, 0) + v\mathbf{P}(u, 1) \quad (9)$$

$$\mathbf{P}(u, v) = \mathbf{P}_1(u, v) + \mathbf{P}_2(u, v) \quad (10)$$

The resulting surface patch described by Equation (10) does not satisfy the boundary conditions.

$$\mathbf{P}(u, 0) = \mathbf{P}(u, 0) + [(1-u)\mathbf{P}(0, 0) + u\mathbf{P}(1, 0)] \quad (11)$$

$$\mathbf{P}(u, 1) = \mathbf{P}(u, 1) + [(1-u)\mathbf{P}(0, 1) + u\mathbf{P}(1, 1)] \quad (12)$$

Let us define a “correction surface” below:

$$\mathbf{P}_3(u, v) = (1-v)[(1-u)\mathbf{P}(0, 0) + u\mathbf{P}(1, 0)] + v[(1-u)\mathbf{P}(0, 1) + u\mathbf{P}(1, 1)] \quad (13)$$

Subtracting $\mathbf{P}_3(u, v)$ from Equation (10) gives

$$\mathbf{P}(u, v) = \mathbf{P}_1(u, v) + \mathbf{P}_2(u, v) - \mathbf{P}_3(u, v) \quad (14)$$

Now $\mathbf{P}(u, v)$ in Equation (14) satisfies the boundary conditions,

$$\mathbf{P}(u, v) = \mathbf{P}_1(u, v) \oplus \mathbf{P}_2(u, v) \quad (15)$$

$$\mathbf{P}(u, v) = - \begin{bmatrix} -1 & (1-u) & u \end{bmatrix} \begin{bmatrix} 0 & \mathbf{P}(u, 0) & \mathbf{P}(u, 1) \\ \mathbf{P}(0, v) & \mathbf{P}(0, 0) & \mathbf{P}(0, 1) \\ \mathbf{P}(1, v) & \mathbf{P}(1, 0) & \mathbf{P}(1, 1) \end{bmatrix} \begin{bmatrix} -1 \\ 1-v \\ v \end{bmatrix} \quad (16)$$

The left column and the upper row of the matrix represent $\mathbf{P}_1(u, v)$ and $\mathbf{P}_2(u, v)$ respectively, while the lower right block represents the correction surface $\mathbf{P}_3(u, v)$.

◇ Assignment 4

Define 4 Bezier curves $\mathbf{P}(0, v)$, $\mathbf{P}(1, v)$, $\mathbf{P}(u, 0)$, $\mathbf{P}(u, 1)$. Write a Matlab program to draw the corresponding bilinearly blended Coon’s patch using Equation (16). Use proper mesh and views to display the surface. Show the Matlab program too. ◇

The functions $-1, 1-u, u, 1-v,$ and v are called blending functions because they blend together four separate boundary curves to give one surface. For cubic blending function , we can define

$$\mathbf{P}_1(u, v) = (2u^3 - 3u^2 + 1)\mathbf{P}(0, v) + (-2u^3 + 3u^2)\mathbf{P}(1, v) \tag{17}$$

$$\mathbf{P}_2(u, v) = (2v^3 - 3v^2 + 1)\mathbf{P}(u, 0) + (-2v^3 + 3v^2)\mathbf{P}(u, 1) \tag{18}$$

$$\mathbf{P}(u, v) = - \begin{bmatrix} -1 & F_1(u) & F_2(u) \end{bmatrix} \begin{bmatrix} 0 & \mathbf{P}(u, 0) & \mathbf{P}(u, 1) \\ \mathbf{P}(0, v) & \mathbf{P}(0, 0) & \mathbf{P}(0, 1) \\ \mathbf{P}(1, v) & \mathbf{P}(1, 0) & \mathbf{P}(1, 1) \end{bmatrix} \begin{bmatrix} -1 \\ F_1(v) \\ F_2(v) \end{bmatrix} \tag{19}$$

4. Triangular Patches

In triangulation techniques, three parameters u, v and w are used and the parametric domain is defined by a symmetric unit triangle of $0 \leq u \leq 1, 0 \leq v \leq 1,$ and $0 \leq w \leq 1,$ as shown in Figure 2. The coordinates u, v and w are called ‘‘barycentric coordinates.’’ While the coordinate w is not independent of u and v (note that $u+v+w=1$ for any point in the domain), it is introduced to emphasize the symmetry properties of the barycentric coordinates.

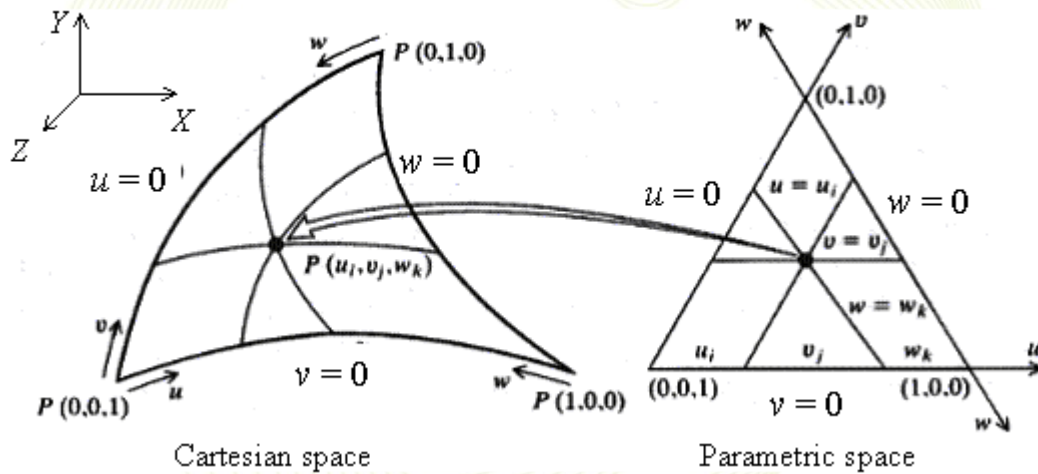


Figure 2. Representation of a triangular patch.

A triangular Bezier patch is defined by

$$\mathbf{P}(u, v, w) = \sum_{i,j,k} \mathbf{P}_{ijk} B_{i,j,k,n}(u, v, w) \quad 0 \leq u \leq 1, 0 \leq v \leq 1, 0 \leq w \leq 1 \tag{20}$$

$$B_{i,j,k,n} = \frac{n!}{i!j!k!} u^i v^j w^k \quad (21)$$

For example, as shown in Figure 3, a cubic triangular patch is

$$\begin{aligned} \mathbf{P}(u, v, w) = & \mathbf{P}_{003} \cdot w^3 + \mathbf{P}_{012} \cdot 3vw^2 + \mathbf{P}_{021} \cdot 3v^2w + \mathbf{P}_{030} \cdot v^3 \\ & + \mathbf{P}_{102} \cdot 3uw^2 + \mathbf{P}_{111} \cdot 6uvw + \mathbf{P}_{120} \cdot 3uv^2 \\ & + \mathbf{P}_{201} \cdot 3u^2w + \mathbf{P}_{210} \cdot 3u^2v \\ & + \mathbf{P}_{300} \cdot u^3 \end{aligned} \quad (22)$$

◇Assignment 5

Define 10 control points in a similar pattern shown in Figure 2. You can also define an xy “ground plane” first, then adjust the z -coordinates of your control points. Write a Matlab program to draw the corresponding cubic triangular patch using Equation (22). Use proper mesh and views to display the surface. Show the Matlab program too. ◇

For $n=4$, the triangular patch is defined as

$$\begin{aligned} \mathbf{P}(u, v, w) = & \mathbf{P}_{004} \cdot w^4 + \mathbf{P}_{013} \cdot 4vw^3 + \mathbf{P}_{022} \cdot 6v^2w^2 + \mathbf{P}_{031} \cdot 4v^3w + \mathbf{P}_{040} \cdot v^4 \\ & + \mathbf{P}_{103} \cdot 4uw^3 + \mathbf{P}_{112} \cdot 12uvw^2 + \mathbf{P}_{121} \cdot 12uv^2w + \mathbf{P}_{130} \cdot 4uv^3 \\ & + \mathbf{P}_{202} \cdot 6u^2w^2 + \mathbf{P}_{211} \cdot 12u^2vw + \mathbf{P}_{220} \cdot 6u^2v^2 \\ & + \mathbf{P}_{301} \cdot 4u^3w + \mathbf{P}_{310} \cdot 4u^3v \\ & + \mathbf{P}_{400} \cdot u^4 \end{aligned} \quad (23)$$

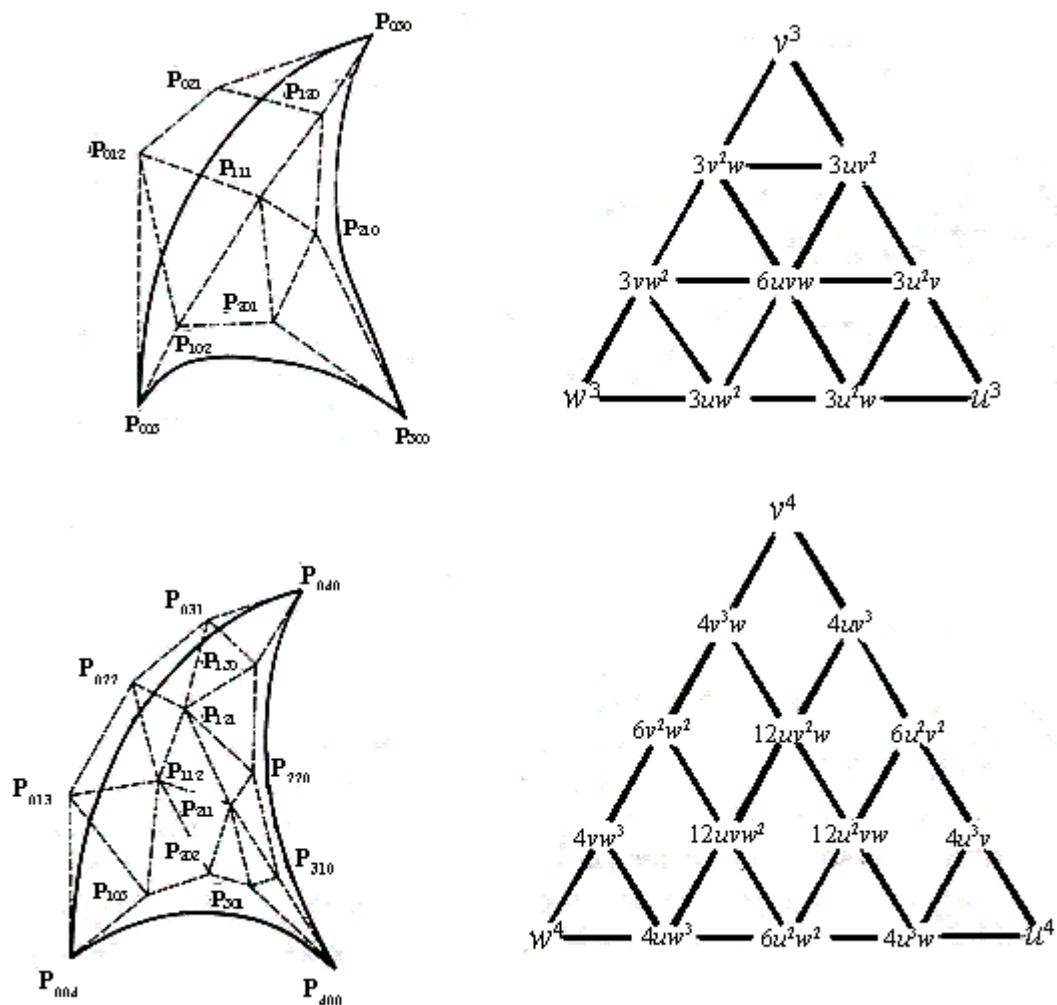


Figure 3. Triangular Bezier patches

◇ Assignment 6

Derive Equation (23) using the definitions in Equation (20) and Equation (21). ◇