MA 323 Geometric Modelling
Course Notes: Day 25
Surface Patches and Rectangular Bezier Patches

David L. Finn

Today, we start looking in depth at surface patches, in particular rectangular Bezier patches. Recall, for our purposes a surface patch is a parametric surface $\xi : [0,1] \times [0,1] \to \mathbb{R}^3$. Typically, we will view a surface patch as a small piece of a larger surface, one square in a *patchwork* quilt. The patch itself can be thought of as a piece of cloth that is woven with thread (curves) in two independent direction u and v, see diagram below, drawn in a wireframe or thread view.

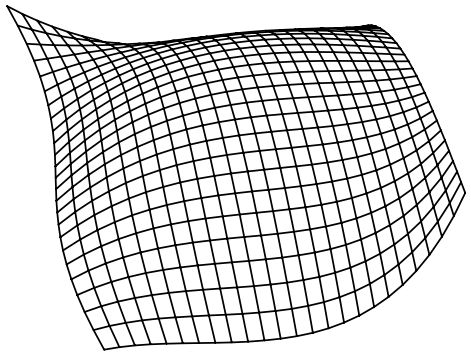

Figure 1: A wireframe rendering of a surface patch

25.1 Bezier Patches

A Bezier patch is a special type of surface patch, defined by given a doubly indexed set of control points $p_{i,j}$, forming a control net to define the individual curves. Think of the double indexing as an integer points in a rectangular grid. Each point in the grid is associated to a control point (see diagram below), and each connecting line is used to shape the surface. We view the surface as a mapping of the rectangular grid into space. However, we note that the algorithm will not map the control points the grid is to represent on to the surface.

The definition of a Bezier patch is as a tensor product. The doubly indexed set of control points are viewed as an $(m+1) \times (n+1) \times 3$ matrix, that is a three dimensional matrix or a tensor. To generate a surface from the control net, we note that each column or row of the control net can generate a Bezier curve. To achieve the surface patch, we apply de Casteljau's algorithm on each row using the variable u, and then n on the remaining points, we apply de Casteljau's algorithm in the variable v, see diagram below. This is the two-step de Casteljau's algorithm, described in more detail below.

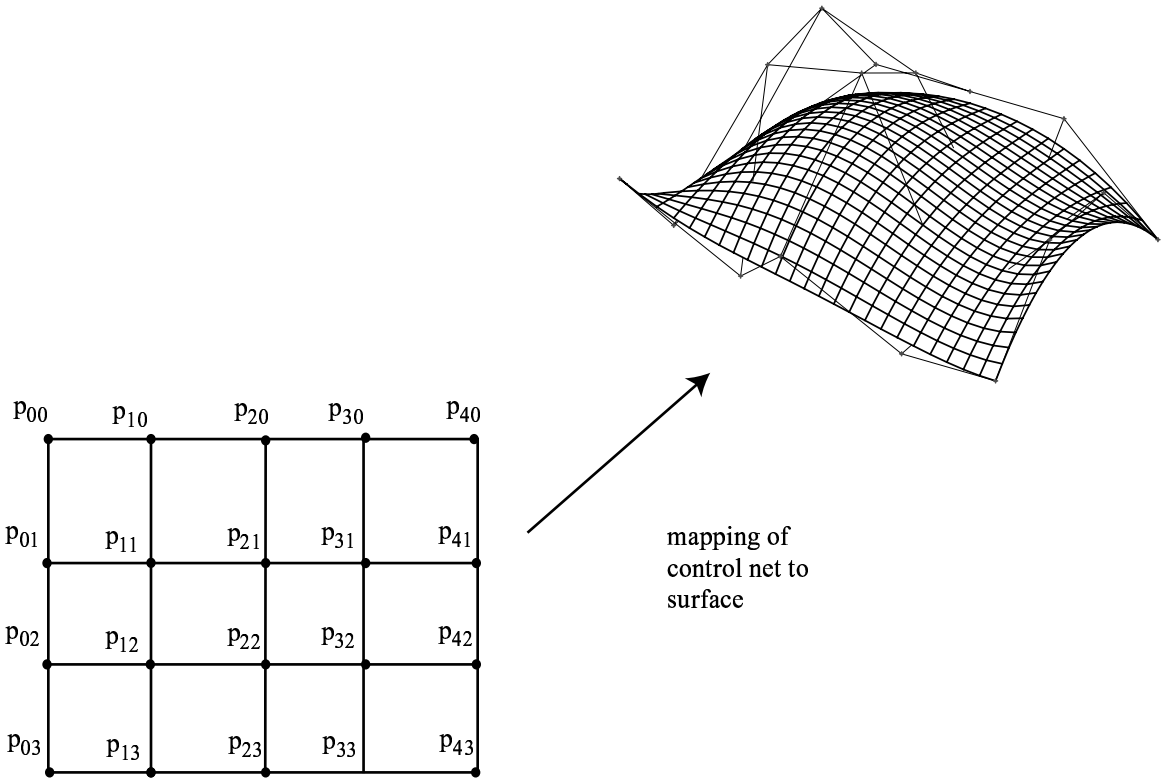

Figure 2: Control Grid and the View of Surface as Mapping Grid to Space

The functional representation of a Bezier patch yields more insight into why it is called a tensor product surface patch. First, we note that the size of the rectangular grid determines the type of Bezier patch. For instance, given the grid is $(m+1) \times (n+1)$ the surface is a polynomial function of degree $m \times n$, meaning the surface is a sum of mth degree polynomials in the variable u times nth degree polynomials in the variable v. The patch is thus functionally described by

$$X(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} B_i^m(u)\, B_j^n(v)\, p_{i,j},$$

which can be written as a quadratic form in terms of matrix or linear algebra as

$$X(u,v) = B(u)^T\, P\, B(v)$$

where P is the *matrix* of control points and $B(u)$ is the column vector of the mth degree Bernstein polynomials in u and $B(v)$ is the column vector of the nth degree Bernstein polynomials in v. Since, P is really a tensor, the Bezier patch are a specific instance of a more general of construction called a tensor product surface.

We will mainly consider patches where $m = n = 1$ bilinear patches and patches where $m = n = 3$ bicubic patches. The first type where $m = n = 1$ is a bilinear patch because in each direction u or v the corresponding curves where one of the variables is held constant is a straight line. The second type where $m = n = 3$ is a bicubic patch because in each direction u or v the corresponding curves where one of the variables is held constant is a

cubic curve. This type of patch is the standard basic type of surface used in geometric modelling.

25.2 Bilinear Bezier Patches

To form a bilinear Bezier patch, you give four points, $p_{0,0}$, $p_{1,0}$, $p_{0,1}$ and $p_{1,1}$. Then, consider the surface formed by the lines joining the points $P_0(u) = (1-u)\,p_{0,0} + u\,p_{1,0}$ and $P_1(u) = (1-u)\,p_{0,1} + u\,p_{1,1}$. Thus, the surface is $B(u,v) = (1-v)\,P_0(u) + v\,P_1(u) = (1-u)(1-v)\,p_{0,0} + u(1-v)\,p_{1,0} + (1-u)v\,p_{0,1} + uv\,p_{1,1}$. Notice, we could also define the surface as formed by the lines joining the points $Q_0(v) = (1-v)\,p_{0,0} + v\,p_{0,1}$ and $Q_1(v) = (1-v)\,p_{1,0} + v\,p_{1,1}$. We remark that it can be shown by purely geometric arguments that $B(u,v)$ is a hyperbolic paraboloid, a quadric surface of the form $z = x^2 - y^2$ or actually a rotation of this type of surface.

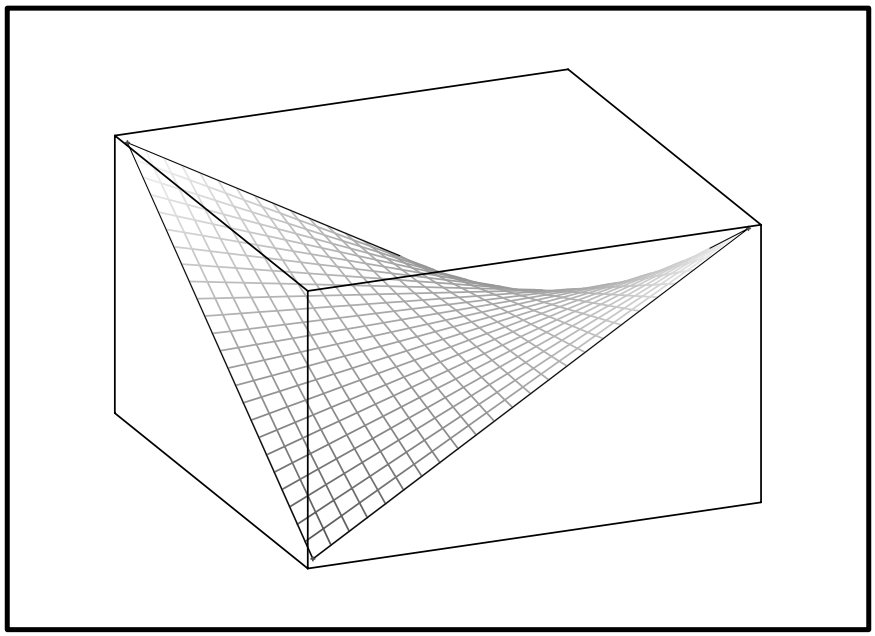

Figure 3: A bilinear Bezier patch

Our purpose in examining bilinear patches is to examine in a simple case how de Casteljau's algorithm works for surfaces. First, notice a bilinear patch is already defined by repeated linear interpolation. In fact, we used de Casteljau's algorithm in defining the curve. We first apply de Castelajau's algorithm in one variable to form control points for a Bezier curve. Then, we apply de Casteljau's algorithm on these control points to define a point on the surface. The important fact in our definition of a bilinear patch is that it does not matter on the order that we use when we apply de Casteljau's algorithm in two variables. We can apply it in u first and then v or in v first then u, and no matter what we get the same point on the curve.

25.3 The two-step de Casteljau's algorithm

First, recall that de Casteljau's algorithm for a curve is given by repeated linear interpolation on the control points p_0, p_1, p_2, $\cdots$, p_n. At each step we define auxiliary control points, then repeat the process on the auxiliary control points. The algorithm is given by picking a number t, and setting $p_i^0 = p_i$ where i ranges from 0 to n. Then, we define a recursion relation for the auxiliary control points p_i^k by

$$p_i^{k+1} = (1-t)\, p_i^k + t\, p_{i+1}^k$$

where i ranges from 0 to $n-k$, and k ranges from 1 to n. The final point p_0^n when the recursion stops is a point on the curve.

To apply de Casteljau's algorithm on a surface, we choose a u and a v, and set $p_{i,j}^{0,0} = p_{i,j}$. Apply de Casteljau's algorithm in u, using the index i. Define $p_{i,j}^{k+1,0} = (1-u)\, p_{i,j}^{k,0} + u\, b_{i+1,j}^{k,0}$ where i ranges from 0 to $m-k$ and k ranges from 1 to m. We do this for each value of j. The points $p_{0,j}^{m,0}$ are control points for a Bezier curve of degree n. We can then apply de Casteljau's algorithm on these control points to get

$$p_{0,j}^{m,l+1} = (1-v)\, p_{0,j}^{m,l} + v\, p_{0,j+1}^{m,l}.$$

The index j ranges from 0 to $n-l$ and l ranges from 1 to n. The final point $p_{0,0}^{m,n}$ lies on the surface. See the diagram below that illustrates the procedure.

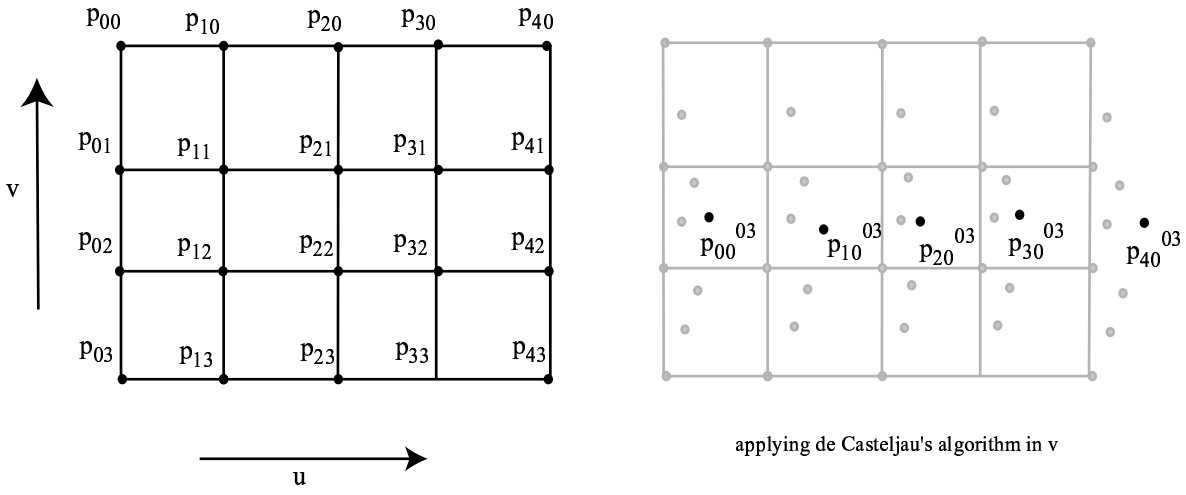

Figure 4: Applying de Casteljau's algorithm in one variable

One question should immediately arise. Does it matter whether we apply de Casteljau's algorithm in u first or v first in the above two step de Casteljau's algorithm? It should not matter. But that needs to be checked. It does not matter.

25.4 One-Step de Casteljau algorithm

We may also use de Casteljau's algorithm in a more surface method. It relies on bilinear interpolation on each set of four control points $p_{i,j}$, $p_{i+1,j}$, $p_{i,j+1}$, $p_{i+1,j+1}$. This method is particularly useful, for bicubic curves or more generally any Bezier surface on an $m \times m$ grid.

The method works, by first choosing a u and a v, and setting $p^0_{i,j} = p_{i,j}$ the given control points. The recursive formula is now

$$p^{k+1}_{i,j} = (1-u)(1-v)\,p^k_{i,j} + (1-u)v\,p^k_{i,j+1} + u(1-v)\,p^k_{i+1,j} + uv\,p^k_{i+1,j+1}$$

where i and j range between 0 and $m-k$ and k ranges from 0 to m. The final point $p^m_{0,0}$ is a point on the surface. $p^m_{0,0} = B(u,v)$. See the diagram below that illustrates the procedure

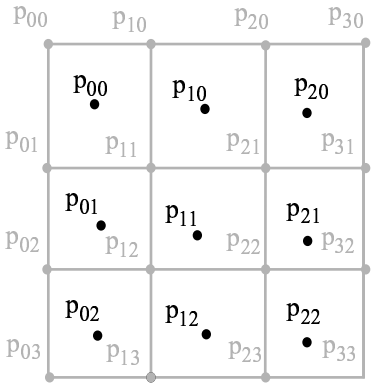

Figure 5: One iteration of the one-step de Casteljau's algorithm

25.5 Properties of Bezier patches

Bezier patches have most of the properties of Bezier curves. There are only a few deviations.

- *Affine invariance.* This is a direct result of de Casteljau's algorithm having affine invariance. We also have that $\sum_{i=0}^{m}\sum_{j=0}^{n} B^m_i(u)\,B^n_j(v) = 1$, so we have a barycentric combination of the control points of a Bezier patch.
- *Convex hull property.* The surface $B(u,v)$ lies within the convex hull of the control points. This is a consequence of the fact that for $0 \le u, v \le 1$, the Bernstein polynomials are nonnegative, so we have a convex combination of the control points.
- The boundary curves are polynomial curves (Bezier curves) whose control points are given by the control points on the boundary of the control net.
- *End point interpolation.* The four corner points of the control net lie on the surface.
- *Pseudolocal control.* Moving one control point $p_{i,j}$ slightly affects the whole surface, but the effect is concentrated around the affected control point, the parameter value $u = i/m$ and $v = j/n$.
- *Symmetry.* Interchanging the variables or rotating the grid (if bicubic more generally $m = n$) does not change the surface. [If $m \neq n$, there is no symmetry. You can not interchange the variables.]
- *Variation diminishing property.* There is no variation diminishing property. No one has determined what the correct idea of variation diminishing should be.

Tomorrow, we will investigate and play with bicubic Bezier patches. Tonight, you need to familiarize yourself with de Casteljau's algorithm for surfaces.

25.6 EXERCISES

1. Apply the two-step de Casteljau algorithm on the control net

$$\begin{bmatrix} p_{0,0} = [1,0,2] & p_{0,1} = [1,2,4] & p_{0,2} = [1,3,2] \\ p_{1,0} = [2,1,0] & p_{1,1} = [2,2,2] & p_{1,2} = [2,4,0] \\ p_{2,0} = [4,1,0] & p_{2,1} = [3,3,2] & p_{2,2} = [4,4,2] \end{bmatrix}$$

   evaluating u first with $u = 1/2$, and then v with $v = 1/2$.

2. Apply the one-step de Casteljau algorithm on the control net

$$\begin{bmatrix} p_{0,0} = [0,0,4] & p_{0,1} = [0,2,4] & p_{0,2} = [0,4,0] \\ p_{1,0} = [2,0,0] & p_{1,1} = [2,2,4] & p_{1,2} = [2,4,0] \\ p_{2,0} = [4,0,0] & p_{2,1} = [4,2,0] & p_{2,2} = [4,4,8] \end{bmatrix}$$

   with $u = 1/4$ and $v = 1/4$.

3. Show that for a bicubic Bezier patch, applying either of the three forms of de Casteljau's algorithm results in the same answer. The three forms are u first then v, or v first then u in the tensor product formulation, or the one-step formulation using repeated bilinear interpolation.

4. Play with the applet for creating Bezier Patches.