Chap 6. Bézier Patches

6.1 Parametric Surfaces

A parametric surface is defined as a map of the real plane into 3-space.

\[ x(u, v) = \begin{bmatrix} f(u, v) \\ g(u, v) \\ h(u, v) \end{bmatrix} \]

Parametric surfaces are more general than bivariate functions of the form \( z = f(x, y) \).

6.2 Bilinear Patches

- Patch: a finite piece of surface.
- We map the unit square \( \{ (u, v) \mid 0 \leq u, v \leq 1 \} \) to a surface patch defined by \( b_0, b_1, b_0, b_1 \).

\[ x(u, v) = [1-u u] \begin{bmatrix} b_{0,0} & b_{0,1} \\ b_{1,0} & b_{1,1} \end{bmatrix} [1-v \ v] : \text{bilinear patch} \]

\[ = (1-u) p^u + u q^v \]

where \( p^u = (1-u) b_{0,0} + u b_{1,0} \)

\[ q^v = (1-v) b_{0,1} + v b_{1,1} \]

or \( x(u, v) = (1-u) p^u + u q^v \)

where \( p^u = (1-u) b_{0,0} + u b_{1,0} \) and \( q^v = (1-v) b_{0,1} + v b_{1,1} \).

- The domain diagonal is mapped to a quadratic curve

\[ d_1(t) = x(t, t) = [1-t \ t] \begin{bmatrix} b_{0,0} & b_{0,1} \\ b_{1,0} & b_{1,1} \end{bmatrix} [1-t \ t] \]

\[ = (1-t)^2 b_{0,0} + 2(1-t) t \left[ \frac{1}{2} b_{0,1} + \frac{1}{2} b_{1,0} \right] + t^2 b_{1,1} \]

This is a quadratic Bézier curve.
6.3 Bézier Patches

0. The bilinear patch can be written as follows:
\[ x(u, v) = \begin{bmatrix} B_0^0(u) & B_1^0(u) \\ B_0^1(u) & B_1^1(u) \end{bmatrix} \begin{bmatrix} 1b_0,0 & 1b_0,1 \\ 1b_1,0 & 1b_1,1 \end{bmatrix} \begin{bmatrix} B_0^0(v) \\ B_1^0(v) \end{bmatrix} \]

0. Generalization
\[ x(u, v) = \begin{bmatrix} B_0^0(u) & \cdots & B_0^n(u) \\ \vdots & \ddots & \vdots \\ B_m^0(u) & \cdots & B_m^n(u) \end{bmatrix} \begin{bmatrix} 1b_{m,0},0 & \cdots & 1b_{m,0},n \\ \vdots & \ddots & \vdots \\ 1b_{m,n},0 & \cdots & 1b_{m,n},n \end{bmatrix} \begin{bmatrix} B_0^0(v) \\ \vdots \\ B_m^0(v) \end{bmatrix} \]
\[ = M^T B N \] (where \( C = M^T B = [c_0, \ldots, c_n] \))

\[ = C N \to \text{the 2-stage explicit evaluation method} \]

0. The elements \( c_0, \ldots, c_n \) of \( C \) do not depend on the parameter value \( u \); they can be reused in computing several points \( x(u, v_1), x(u, v_2), \ldots \).

0. \( c_0, \ldots, c_n \) are the Bézier control points for the curve \( C U \) with constant \( u \) and variable \( v \). \( \Rightarrow \text{Isoparametric curve} \).

0. Let \( x(u, v) = M^T I D \) where \( I D = B N \)
\[ \Rightarrow \text{The result is the same} \]

---

**Sketch 42.**
Evaluation of a Bézier patch via a \( u = \text{constant iso} \text{curve} \).

**Sketch 43.**
Evaluation of a Bézier patch via a \( v = \text{constant iso} \text{curve} \).
### 6.4 Properties of Bézier Patches

Bézier patches have many properties that are essentially carbon copies of the curve ones.

1. **Endpoint interpolation:** Analogous to the curve case, the patch passes through the four corner control points, that is
   \[ x(0,0) = b_{0,0} \quad x(1,0) = b_{m,0} \]
   \[ x(0,1) = b_{0,n} \quad x(1,1) = b_{m,n}. \]

   However, this property is more powerful for the surface case than for the curve case. We also have that control polygon boundaries are the control points of the patch boundary curves. For example: The curve \( x(u,1) \) has the control polygon \( b_{0,n}, \ldots, b_{m,n} \).

2. **Symmetry:** We could re-index the control net so that any of the corners corresponds to \( b_{0,0} \), and evaluation would result in a patch with the same shape as the original one.

3. **Affine invariance:** Apply an affine map to the control net, and then evaluate the patch. This surface will be identical to the surface created by applying the same affine map to the original patch.

4. **Convex hull property:** For \( (u,v) \in [0,1] \times [0,1] \), the patch \( x(u,v) \) is in the convex hull of the control net.

5. **Bilinear precision:** Sketch 44 illustrates a degree \( m \times n \) patch with boundary control points which are evenly spaced on lines connecting the corner control points, and the interior control points are evenly-spaced on lines connecting boundary control points on adjacent edges. This patch is identical to the bilinear interpolant to the four corner control points.

6. **Tensor product:** Bézier patches are in the class of tensor product surfaces. This property allows Bézier patches to be dealt with in terms of isoparametric curves, which in turn simplifies evaluation and other operations. The breakdown of (6.8) into (6.9) and (6.10) illustrates this point nicely.

The tensor product property is a very powerful conceptual tool for understanding Bézier patches. Sketch 45 illustrates how the shape of a Bézier patch can be thought of as a record of the shape of a template moving through space. This template can change shape as it moves, and its shape and position is guided by “columns” of Bézier control points.
6.5 Derivatives

0. The u-partial derivative:
\[ x_u(u,v) = \frac{\partial x(u,v)}{\partial u} = \frac{\partial x(u)}{\partial u} \]

where \( \Delta^0 \alpha \equiv \alpha \)

0. The v-partial derivative:
\[ x_v(u,v) = \frac{\partial x(u,v)}{\partial v} = \frac{\partial x(v)}{\partial v} \]

where \( \Delta^0 \beta \equiv \beta \)

6.6 Higher Order Derivatives

0. The (k)-th order partial derivative:
\[ x_{\alpha_k}(u,v) = \frac{n!}{(n-k)!} \left( \frac{\partial \Delta^0 \alpha_k}{\partial u} \right) \]

where \( \Delta^0 \alpha_k \equiv \alpha_k \)

0. The (k)-th order partial derivative:
\[ x_{\alpha_k}(u,v) = \frac{n!}{(n-k)!} \left( \frac{\partial \Delta^0 \alpha_k}{\partial v} \right) \]

where \( \Delta^0 \beta_k \equiv \beta_k \)

0. Note that \( B_0^n(0) = 1, B_i^n(0) = 0, i = 1, 2, \ldots, n \),
\( B_n^n(1) = 1, B_i^n(1) = 0, i = 0, \ldots, n-1 \).

\[ x_{uv}(0,0) = mn \Delta^1 \beta_0, \quad x_{uv}(0,1) = mn \Delta^1 \beta_0, \quad x_{uv}(1,0) = mn \Delta^1 \beta_{m+1}, \quad x_{uv}(1,1) = mn \Delta^1 \beta_{m+1} \]
6.7 The de Casteljau Algorithm

0. In computing $C = MTB$, we may use the de Casteljau algorithm for each of the $C_i$. Moreover, in the final evaluation step for $\mathbf{x}(u,v) = C_N$, we may also use the de Casteljau algorithm. $\Rightarrow$ the 2-stage de Casteljau evaluation method.

0. The advantage is that it allows computation of a derivative along with computation of a point. Once we have $C$, we evaluate point $\mathbf{x}(u,v)$ and tangent $\mathbf{x}_u$.

0. We may first compute $\mathbf{ID} = IBN$ and then $\mathbf{x} = MTID$. The tangent to the curve with control polygon ID is $\mathbf{x}_u$.

6.8 Normals

0. The normal vector is perpendicular to the surface at $\mathbf{x}$.

0. The tangent plane of a surface at $\mathbf{x}$ is defined by $\mathbf{x}, \mathbf{x}_u, \mathbf{x}_v$, a point and two vectors.

0. The normal $\mathbf{n}$ is a unit vector: $\mathbf{n} = \frac{\mathbf{x}_u \times \mathbf{x}_v}{\| \mathbf{x}_u \times \mathbf{x}_v \|}$.

0. Caution: $\| \mathbf{x}_u \times \mathbf{x}_v \|$ should not be zero!

But, in some degenerate cases, it can be zero. Thus, a zero division check is a good idea.

0. As presented in Section 6.7, computing $\mathbf{x}_u$ and $\mathbf{x}_v$ would require two patch evaluations. However, there is a trick to evaluate $\mathbf{x}, \mathbf{x}_u, \mathbf{x}_v$ almost simultaneously.

1. Compute $(n-1)$ levels of the de Casteljau algorithm for all $(m+1)$ rows of control points; these are w.r.t $N$.
2. Compute $(m-1)$ levels of the alg. for each of them w.r.t $U$.
3. Now, four points define a bilinear patch.
The tangent plane at \((u,v)\) agrees with the surface's tangent plane at \((u,v)\). Thus, we evaluate the bilinear patch and compute its partials at \((u,v)\).

(The 3-stage de Casteljau evaluation method: a slightly modified version of the 2-stage de Casteljau plus a bilinear patch evaluation.)

![Sketch 52. A schematic description of the 3-stage algorithm.](image)

![Sketch 53. Computation of a normal vector.](image)

6.9 Changing Degrees

0. A Bézier patch has two degrees: \(m\) in the \(u\)-direction and \(n\) in the \(v\)-direction. Each may be increased.

0. To raise \(m\) to \(m+1\): There are \((m+1)\) columns of control points, each column containing \((m+2)\) control points. These columns are obtained from the original columns by the process of degree elevation for curves.

6.10 Subdivision

0. A patch may be subdivided into two patches along a \(u_0\)-parameter curve. ⇒ Perform curve subdivision for each degree \(m\) column of the control net at parameter \(u_0\).
6.11 Ruled Bézier Patches

0. A ruled Bézier patch is linear in one isoparametric direction

\[
\begin{align*}
\mathbf{x}(u, v) &= (1-v)\mathbf{x}(u, 0) + v\mathbf{x}(u, 1) \quad \text{if the v-direction is linear} \\
\mathbf{x}(u, v) &= (1-u)\mathbf{x}(0, v) + u\mathbf{x}(1, v) \quad \text{if the u-direction}
\end{align*}
\]

0. The control points of a ruled Bézier patch are

- the control points of the two given Bézier curves.

⇒ The two curves should be of the same degree, which can always be achieved through degree elevation.

\[
\begin{bmatrix}
\mathbf{x}(u, v) \\
\mathbf{y}(u, v) \\
\end{bmatrix} =
\begin{bmatrix}
B_0^m(u) & \cdots & B_m^m(u) \\
\vdots & \ddots & \vdots \\
B_0^m(u) & \cdots & B_m^m(u)
\end{bmatrix}
\begin{bmatrix}
\mathbf{b}_0, v \\
\vdots \\
\mathbf{b}_m, v
\end{bmatrix}
\]

0. A developable surface is a special ruled surface.

This type of surface is important in manufacturing because it can be "developed" by bending a sheet metal, without tearing or stretching the metal.

0. Special conditions exist for a ruled surface to be developable. (The Gaussian curvature must be zero everywhere; see the definition in Section 6.3.)
6.12 Functional Bézier Patches

1. The graph of a functional surface can be thought of as a parametric surface of the form:

\[
\begin{bmatrix}
  x \\
  y \\
  z
\end{bmatrix} = \begin{bmatrix}
  x(u) \\
  y(v) \\
  z(u, v)
\end{bmatrix} = \begin{bmatrix}
  u \\
  v \\
  f(u, v)
\end{bmatrix}
\]

2. Functional Bézier patches are single-valued. This makes these surfaces quite useful in manufacturing via stamping sheet metal.

3. A functional Bézier patch defined over \([0,1] \times [0,1]\) has its control points of the form \(b_{ij}\):

\[
b_{ij} = \begin{bmatrix}
  i/m \\
  j/n \\
  bi,j
\end{bmatrix}
\]

6.13 Monomial Patches

4. A monomial patch is defined as

\[
X(u, v) = \begin{bmatrix}
  1 & u & \cdots & u^m
\end{bmatrix} \begin{bmatrix}
  a_{0,0} & \cdots & a_{0,n} \\
  \vdots & \ddots & \vdots \\
  a_{m,0} & \cdots & a_{m,n}
\end{bmatrix} \begin{bmatrix}
  u \\
  v \\
  \vdots \\
  v^n
\end{bmatrix}
\]

or

\[
X(u, v) = U^T A V.
\]

5. \(a_{0,0}\) is a point \(X(0,0)\) and all other \(a_{i,j}\): partial derivatives.

6. Conversion between surfaces:

\[
a_{i,j} = \binom{m}{i} \binom{n}{j} \Delta_i^j \ b_{0,0,0}.
\]