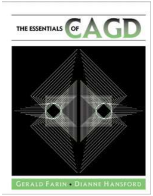
The Essentials of CAGD

Chapter 6: Bézier Patches

Gerald Farin & Dianne Hansford

CRC Press, Taylor & Francis Group, An A K Peters Book
www.farinhansford.com/books/essentials-cagd

©2000



Outline

Introduction to Bézier Patches

The “Utah” teapot composed of Bézier patches

Surfaces:

– Basic definitions

– Extend the concept of Bézier curves

Parametric Surfaces

Parametric curve: mapping of the real line into 2- or 3-space

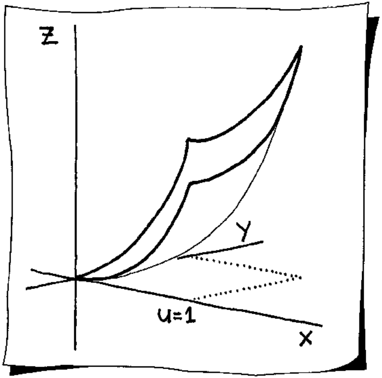
Parametric surface: mapping of the real plane into 3-space

$\mathbb{R}^2$ is the domain of the surface
– A plane with a (u, v) coordinate system

Corresponding 3D surface point:

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

Parametric Surfaces



Example:
Parametric surface

$$\mathbf{x}(u, v) = \begin{bmatrix} u \\ v \\ u^2 + v^2 \end{bmatrix}$$

Only a portion of surface illustrated

This is a functional surface

Parametric surfaces may be rotated or moved around
– More general than $z = f(x, y)$

Bilinear Patches

Typically interested in a finite piece of a parametric surface
- The image of a rectangle in the domain

The finite piece of surface called a patch

Let domain be the *unit square*

$$\{(u, v) : 0 \leq u, v \leq 1\}$$

Map it to a surface patch defined by four points

$$\mathbf{x}(u, v) = \begin{bmatrix} 1-u & u \end{bmatrix} \begin{bmatrix} \mathbf{b}_{0,0} & \mathbf{b}_{0,1} \\ \mathbf{b}_{1,0} & \mathbf{b}_{1,1} \end{bmatrix} \begin{bmatrix} 1-v \\ v \end{bmatrix}$$

Surface patch is linear in both the u and v parameters
$\Rightarrow$ *bilinear patch*

Bilinear Patches

Bilinear patch:

$$\mathbf{x}(u,v) = \begin{bmatrix} 1-u & u \end{bmatrix} \begin{bmatrix} \mathbf{b}_{0,0} & \mathbf{b}_{0,1} \\ \mathbf{b}_{1,0} & \mathbf{b}_{1,1} \end{bmatrix} \begin{bmatrix} 1-v \\ v \end{bmatrix}$$

Geometric interpretation: rewrite as

$$\mathbf{x}(u,v) = (1-v)\mathbf{p}^u + v\mathbf{q}^u$$

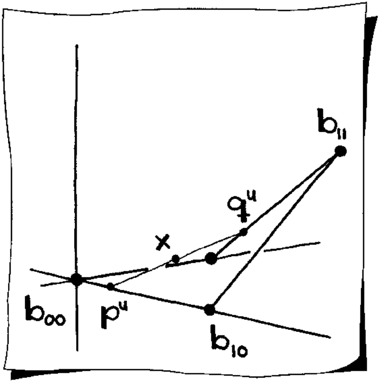
where

$$\mathbf{p}^u = (1-u)\mathbf{b}_{0,0} + u\mathbf{b}_{1,0}$$
$$\mathbf{q}^u = (1-u)\mathbf{b}_{0,1} + u\mathbf{b}_{1,1}$$

Bilinear Patches

Example: Given four points $\mathbf{b}_{i,j}$ and compute $\mathbf{x}(0.25, 0.5)$

$$\mathbf{b}_{0,0} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{b}_{1,0} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{b}_{0,1} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \mathbf{b}_{1,1} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$



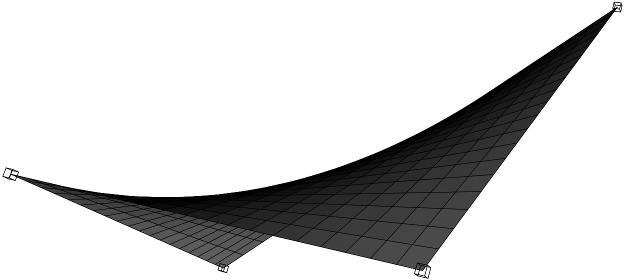
$$\mathbf{p}^u = 0.75 \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + 0.25 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.25 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{q}^u = 0.75 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + 0.25 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.25 \\ 1 \\ 0.25 \end{bmatrix}$$

$$\mathbf{x}(0.25, 0.5) = 0.5\mathbf{p}^u + 0.5\mathbf{q}^u = \begin{bmatrix} 0.25 \\ 0.5 \\ 0.125 \end{bmatrix}$$

Bilinear Patches

Rendered image of patch in previous example



Bilinear Patches

Bilinear patch:

$$\mathbf{x}(u,v) = (1-v)\mathbf{p}^u + v\mathbf{q}^u$$

Is equivalent to

$$\mathbf{x}(u,v) = (1-u)\mathbf{p}^v + u\mathbf{q}^v$$

where

$$\mathbf{p}^v = (1-v)\mathbf{b}_{0,0} + v\mathbf{b}_{0,1}$$
$$\mathbf{q}^v = (1-v)\mathbf{b}_{1,0} + v\mathbf{b}_{1,1}$$

Bilinear Patches

Bilinear patch also called a hyperbolic paraboloid

Isoparametric curve: only one parameter is allowed to vary

Isoparametric curves on a bilinear patch $\Rightarrow$ 2 families of straight lines
$(\bar{u}, v)$: line constant in u but varying in v
$(u, \bar{v})$: line constant in v but varying in u

Four special isoparametric curves (lines):

$$(u, 0) \qquad (u, 1) \qquad (0, v) \qquad (1, v)$$

Bilinear Patches

A hyperbolic paraboloid also contains *curves*

Consider the line $u = v$ in the domain

As a parametric line: $u(t) = t$, $v(t) = t$

Domain diagonal mapped to the 3D curve on the surface

$$\mathbf{d}(t) = \mathbf{x}(t, t)$$

In more detail:

$$\mathbf{d}(t) = \begin{bmatrix} 1-t & t \end{bmatrix} \begin{bmatrix} \mathbf{b}_{0,0} & \mathbf{b}_{0,1} \\ \mathbf{b}_{1,0} & \mathbf{b}_{1,1} \end{bmatrix} \begin{bmatrix} 1-t \\ t \end{bmatrix}$$

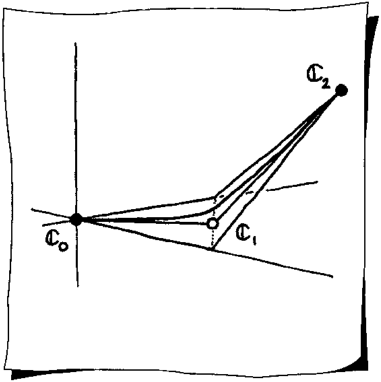
Collecting terms now gives

$$\mathbf{d}(t) = (1-t)^2\mathbf{b}_{0,0} + 2(1-t)t[\frac{1}{2}\mathbf{b}_{0,1} + \frac{1}{2}\mathbf{b}_{1,0}] + t^2\mathbf{b}_{1,1}$$

$\Rightarrow$ quadratic Bézier curve

Bilinear Patches

Example: Compute the curve on the surface for $u(t) = t$, $v(t) = t$



$$\mathbf{c}_0 = \mathbf{b}_{0,0} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{c}_1 = \frac{1}{2}[\mathbf{b}_{1,0} + \mathbf{b}_{0,1}] = \begin{bmatrix} 0.5 \\ 0.5 \\ 0 \end{bmatrix}$$

$$\mathbf{c}_2 = \mathbf{b}_{1,1} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\mathbf{d}(t) = \mathbf{c}_0 B_0^2(t) + \mathbf{c}_1 B_1^2(t) + \mathbf{c}_2 B_2^2(t)$$

Bézier Patches

Bilinear patch using linear Bernstein polynomials:

$$\mathbf{x}(u,v) = \begin{bmatrix} B_0^1(u) & B_1^1(u) \end{bmatrix} \begin{bmatrix} \mathbf{b}_{0,0} & \mathbf{b}_{0,1} \\ \mathbf{b}_{1,0} & \mathbf{b}_{1,1} \end{bmatrix} \begin{bmatrix} B_0^1(v) \\ B_1^1(v) \end{bmatrix}$$

Generalization:

$$\mathbf{x}(u,v) = \begin{bmatrix} B_0^m(u) & \dots & B_m^m(u) \end{bmatrix} \begin{bmatrix} \mathbf{b}_{0,0} & \dots & \mathbf{b}_{0,n} \\ \vdots & & \vdots \\ \mathbf{b}_{m,0} & \dots & \mathbf{b}_{m,n} \end{bmatrix} \begin{bmatrix} B_0^n(v) \\ \vdots \\ B_n^n(v) \end{bmatrix}$$

$$= \mathbf{b}_{0,0} B_0^m(u) B_0^n(v) + \ldots + \mathbf{b}_{i,j} B_i^m(u) B_j^n(v) + \ldots + \mathbf{b}_{m,n} B_m^m(u) B_n^n(v)$$

Examples: $m = n = 1$: bilinear $m = n = 3$: *bicubic*

Bézier Patches

$$\mathbf{x}(u,v) = \begin{bmatrix} B_0^m(u) & \dots & B_m^m(u) \end{bmatrix} \begin{bmatrix} \mathbf{b}_{0,0} & \dots & \mathbf{b}_{0,n} \\ \vdots & & \vdots \\ \mathbf{b}_{m,0} & \dots & \mathbf{b}_{m,n} \end{bmatrix} \begin{bmatrix} B_0^n(v) \\ \vdots \\ B_n^n(v) \end{bmatrix}$$

Abbreviated as

$$\mathbf{x}(u,v) = M^{\mathrm{T}}\mathbf{B}N$$

2-stage explicit evaluation method at given (u, v)
Step 1: generate $\mathbf{c}_i$

$$\mathbf{C} = M^{\mathrm{T}}\mathbf{B} = [\mathbf{c}_0, \dots, \mathbf{c}_n]$$

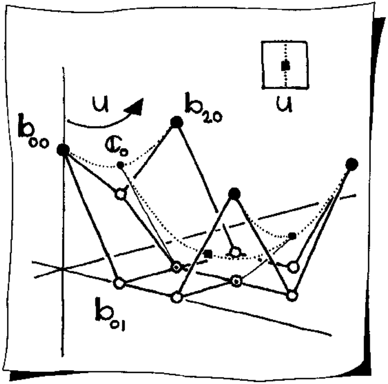
Step 2: generate point on surface

$$\mathbf{x}(u,v) = \mathbf{C}N$$

("explicit" because Bernstein polynomials evaluated)

Bézier Patches

$$\mathbf{x}(u,v) = M^{\mathrm{T}}\mathbf{B}N \qquad \Rightarrow \qquad \mathbf{x}(u,v) = \mathbf{C}N$$



Control points $\mathbf{c}_0, \ldots, \mathbf{c}_n$ of $\mathbf{C}$ do not depend on the parameter value v

Curve $\mathbf{C}N$: curve on surface
– Constant u
– Variable v
$\Rightarrow$ isoparametric curve or isocurve

Bézier Patches

Example: Evaluate the 2×3 control net at $(u, v) = (0.5, 0.5)$

$$\mathbf{B} = \begin{bmatrix} \begin{bmatrix} 0\\0\\6 \end{bmatrix} & \begin{bmatrix} 3\\0\\0 \end{bmatrix} & \begin{bmatrix} 6\\0\\0 \end{bmatrix} & \begin{bmatrix} 9\\0\\6 \end{bmatrix} \\ \begin{bmatrix} 0\\3\\3 \end{bmatrix} & \begin{bmatrix} 3\\3\\0 \end{bmatrix} & \begin{bmatrix} 6\\3\\0 \end{bmatrix} & \begin{bmatrix} 9\\3\\0 \end{bmatrix} \\ \begin{bmatrix} 0\\6\\6 \end{bmatrix} & \begin{bmatrix} 3\\6\\0 \end{bmatrix} & \begin{bmatrix} 6\\6\\0 \end{bmatrix} & \begin{bmatrix} 9\\6\\6 \end{bmatrix} \end{bmatrix}$$

Step 1) Compute quadratic Bernstein polynomials for $u = 0.5$:

$$M^{\mathrm{T}} = \begin{bmatrix} 0.25 & 0.5 & 0.25 \end{bmatrix}$$

$\Rightarrow$ Intermediate control points

$$\mathbf{C} = M^{\mathrm{T}}\mathbf{B} = \begin{bmatrix} \begin{bmatrix} 0\\3\\4.5 \end{bmatrix} & \begin{bmatrix} 3\\3\\0 \end{bmatrix} & \begin{bmatrix} 6\\3\\0 \end{bmatrix} & \begin{bmatrix} 9\\3\\3 \end{bmatrix} \end{bmatrix}$$

Bézier points of an isoparametric curve containing $\mathbf{x}(0.5, 0.5)$

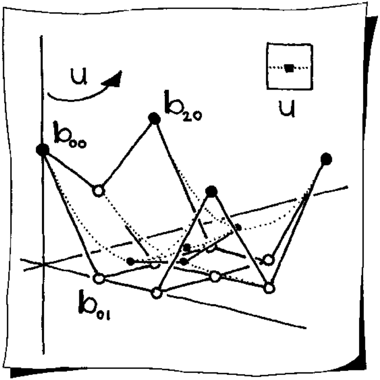
Bézier Patches

Step 2) Compute cubic Bernstein polynomials for $v = 0.5$:

$$N = \begin{bmatrix} 0.125 \\ 0.375 \\ 0.375 \\ 0.125 \end{bmatrix}$$

$$\mathbf{x}(0.5, 0.5) = \mathbf{C}N = \begin{bmatrix} 4.5 \\ 3 \\ 0.9375 \end{bmatrix}$$

Bézier Patches



v-isoparametric curve

Another approach to
2-stage explicit evaluation:

$$\mathbf{x}(u,v) = M^{\mathrm{T}}\mathbf{B}N$$

$$\mathbf{D} = \mathbf{B}N$$

$$\mathbf{x} = M^{\mathrm{T}}\mathbf{D}$$

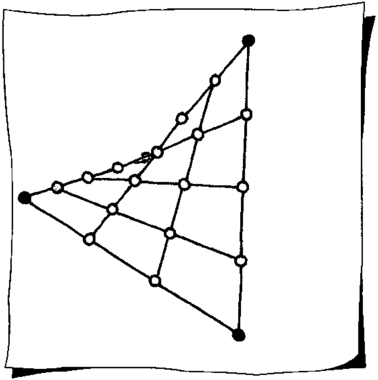
Properties of Bézier Patches

Bézier patches properties essentially the same as the curve ones

1. Endpoint interpolation:
   – Patch passes through the four corner control points
   – Control polygon boundaries define patch boundary curves
2. Symmetry:
   Shape of patch independent of corner selected to be $\mathbf{b}_{0,0}$
3. Affine invariance:
   Apply affine map to control net and then evaluate
   identical to applying affine map to the original patch
4. Convex hull property:
   $\mathbf{x}(u, v)$ in the convex hull of the control net for $(u, v) \in [0, 1] \times [0, 1]$
5. Bilinear precision: Sketch on next slide
6. Tensor product:
   $\Rightarrow$ evaluation via isoparametric curves

Properties of Bézier Patches

A degree 3×4 control net with bilinear precision

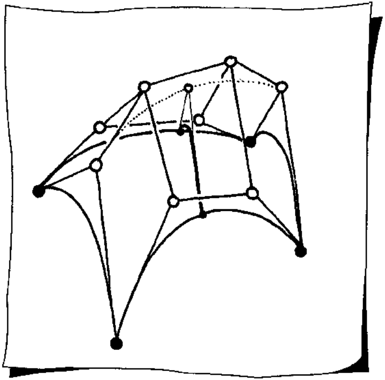


Boundary control points evenly spaced on lines connecting the corner control points

Interior control points evenly-spaced on lines connecting boundary control points on adjacent edges

Properties of Bézier Patches

Tensor product property very powerful conceptual tool for understanding Bézier patches

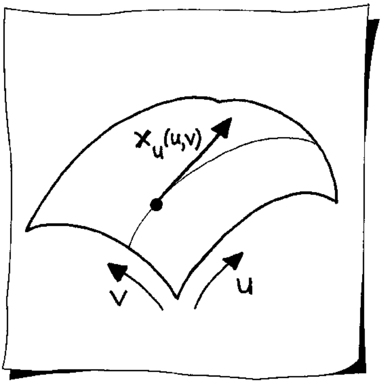


Shape as a record of the shape of a template moving through space

Template can change shape as it moves
Shape and position is guided by “columns” of Bézier control points

Derivatives

A derivative is the tangent vector of a curve on the surface
Called a partial derivative



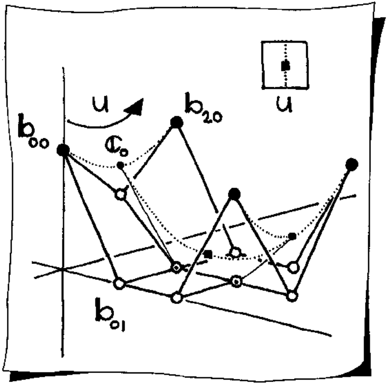
There are two isoparametric curves through a surface point

The $v = constant$ curve
is a curve on the surface with parameter u
– Differentiate with respect to u

$$\mathbf{x}_u(u,v) = \frac{\partial \mathbf{x}(u,v)}{\partial u}$$

Called the $u-$partial

Derivatives



Control polygon C for the $u = 0.5$ isoparametric curve

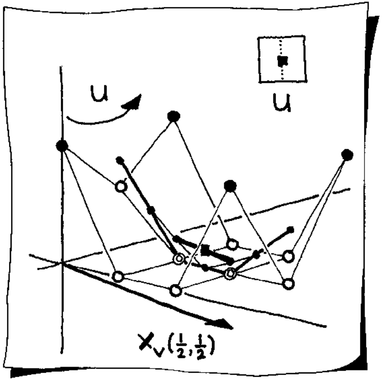
Example: Find partial $\mathbf{x}_v(0.5, 0.5)$ of

$$\mathbf{B} = \begin{bmatrix} \begin{bmatrix}0\\0\\6\end{bmatrix} & \begin{bmatrix}3\\0\\0\end{bmatrix} & \begin{bmatrix}6\\0\\0\end{bmatrix} & \begin{bmatrix}9\\0\\6\end{bmatrix} \\ \begin{bmatrix}0\\3\\3\end{bmatrix} & \begin{bmatrix}3\\3\\0\end{bmatrix} & \begin{bmatrix}6\\3\\0\end{bmatrix} & \begin{bmatrix}9\\3\\0\end{bmatrix} \\ \begin{bmatrix}0\\6\\6\end{bmatrix} & \begin{bmatrix}3\\6\\0\end{bmatrix} & \begin{bmatrix}6\\6\\0\end{bmatrix} & \begin{bmatrix}9\\6\\6\end{bmatrix} \end{bmatrix}$$

Derivatives

Example con't: Derivative curve

$$\mathbf{x}_v(0.5, v) = 3(\Delta\mathbf{c}_0 B_0^2(v) + \Delta\mathbf{c}_1 B_1^2(v) + \Delta\mathbf{c}_2 B_2^2(v))$$



$$\Delta\mathbf{c}_0 = \begin{bmatrix} 3 \\ 0 \\ -4.5 \end{bmatrix} \quad \Delta\mathbf{c}_1 = \begin{bmatrix} 3 \\ 0 \\ 0 \end{bmatrix} \quad \Delta\mathbf{c}_2 \begin{bmatrix} 3 \\ 0 \\ 3 \end{bmatrix}$$

Evaluate at $v = 0.5$

$$\mathbf{x}_v(0.5, 0.5) = \begin{bmatrix} 9 \\ 0 \\ -1.125 \end{bmatrix}$$

$u-$partials $\Rightarrow$ differentiate the isoparametric curve with control points $\mathbf{D} = \mathbf{B}N$

Derivatives

Computing derivatives via a closed-form expression

$$\mathbf{x}_u(u,v) = m \begin{bmatrix} B_0^{m-1}(u) & \dots & B_{m-1}^{m-1}(u) \end{bmatrix} \begin{bmatrix} \Delta^{1,0}\mathbf{b}_{0,0} & \dots & \Delta^{1,0}\mathbf{b}_{0,n} \\ \vdots & & \vdots \\ \Delta^{1,0}\mathbf{b}_{m-1,0} & \dots & \Delta^{1,0}\mathbf{b}_{m-1,n} \end{bmatrix} \begin{bmatrix} B_0^n(v) \\ \vdots \\ B_n^n(v) \end{bmatrix}$$

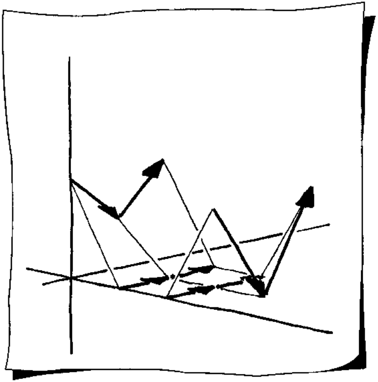
$\Delta^{1,0}\mathbf{b}_{i,j}$ denote forward differences:

$$\Delta^{1,0}\mathbf{b}_{i,j} = \mathbf{b}_{i+1,j} - \mathbf{b}_{i,j}$$

$\Rightarrow$ Closed-form u-partial derivative expression is a degree $(m-1) \times n$ patch with a control net consisting of vectors rather than points

Derivatives

u-partial formed from differences of control points of the original patch in the u-direction



$$\mathbf{x}_u(u,v) = 2\begin{bmatrix} B_0^1(u) & B_1^1(u) \end{bmatrix} \mathbf{B}' \begin{bmatrix} B_0^3(v) \\ B_1^3(v) \\ B_2^3(v) \\ B_3^3(v) \end{bmatrix}$$

$$\mathbf{B}' = \begin{bmatrix} \begin{bmatrix} 0 \\ 3 \\ -3 \end{bmatrix} & \begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 3 \\ -6 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 3 \\ 3 \end{bmatrix} & \begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 3 \\ 6 \end{bmatrix} \end{bmatrix}$$

$$\mathbf{x}_u(0.5, 0.5) = \begin{bmatrix} 0 \\ 6 \\ 0 \end{bmatrix}$$

Derivatives

Closed-form v-partial derivative

$$\mathbf{x}_v(u,v) = n \begin{bmatrix} B_0^m(u) & \dots & B_m^m(u) \end{bmatrix} \begin{bmatrix} \Delta^{0,1}\mathbf{b}_{0,0} & \dots & \Delta^{0,1}\mathbf{b}_{0,n-1} \\ \vdots & & \vdots \\ \Delta^{0,1}\mathbf{b}_{m,0} & \dots & \Delta^{0,1}\mathbf{b}_{m,n-1} \end{bmatrix} \begin{bmatrix} B_0^{n-1}(v) \\ \vdots \\ B_{n-1}^{n-1}(v) \end{bmatrix}$$

$$\Delta^{0,1}\mathbf{b}_{i,j} = \mathbf{b}_{i,j+1} - \mathbf{b}_{i,j}$$

$\Rightarrow$ Closed-form v-partial derivative is a degree $m \times (n-1)$ patch

Higher Order Derivatives

A Bézier patch may be differentiated several times
$\Rightarrow$ Derivatives of order k or k^{th} partials

$v-$partials: $\mathbf{x}_v^{(k)}(u,v) =$

$$\frac{n!}{(n-k)!}\begin{bmatrix} B_0^m(u) & \dots & B_m^m(u) \end{bmatrix} \begin{bmatrix} \Delta^{0,k}\mathbf{b}_{0,0} & \dots & \Delta^{0,k}\mathbf{b}_{0,n-1} \\ \vdots & & \vdots \\ \Delta^{0,k}\mathbf{b}_{m,0} & \dots & \Delta^{0,k}\mathbf{b}_{m,n-1} \end{bmatrix} \begin{bmatrix} B_0^{n-k}(v) \\ \vdots \\ B_{n-1}^{n-k}(v) \end{bmatrix}$$

k^{th} forward differences $\Delta^{0,k}\mathbf{b}_{i,j}$
– Acting only on the second subscripts

Higher Order Derivatives

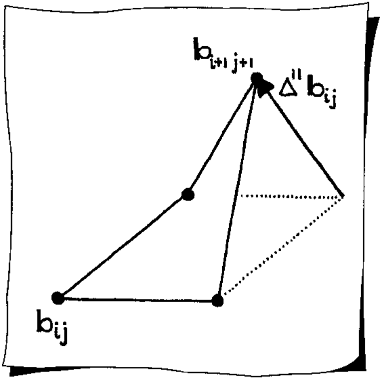
Mixed partial or twist vector

$$\mathbf{x}_{u,v}(u,v) = \frac{\partial \mathbf{x}_u(u,v)}{\partial v} \quad \text{or} \quad \frac{\partial \mathbf{x}_v(u,v)}{\partial u}$$

$\mathbf{x}_{u,v}(u,v) =$

$$mn \begin{bmatrix} B_0^{m-1}(u) & \dots & B_{m-1}^{m-1}(u) \end{bmatrix} \begin{bmatrix} \Delta^{1,1}\mathbf{b}_{0,0} & \dots & \Delta^{1,1}\mathbf{b}_{0,n-1} \\ \vdots & & \vdots \\ \Delta^{1,1}\mathbf{b}_{m-1,0} & \dots & \Delta^{1,1}\mathbf{b}_{m-1,n-1} \end{bmatrix} \begin{bmatrix} B_0^{n-1}(v) \\ \vdots \\ B_{n-1}^{n-1}(v) \end{bmatrix}$$

Higher Order Derivatives



$$
\begin{aligned}
\Delta^{1,1}\mathbf{b}_{i,j} &= \Delta^{0,1}(\mathbf{b}_{i+1,j} - \mathbf{b}_{i,j}) \\
&= \Delta^{0,1}\mathbf{b}_{i+1,j} - \Delta^{0,1}\mathbf{b}_{i,j} \\
&= \mathbf{b}_{i+1,j+1} - \mathbf{b}_{i+1,j} - \mathbf{b}_{i,j+1} + \mathbf{b}_{i,j}
\end{aligned}
$$

Higher Order Derivatives

Example: Bilinear patch

$$\mathbf{b}_{0,0} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{b}_{1,0} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{b}_{0,1} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \mathbf{b}_{1,1} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\begin{aligned} \mathbf{x}_{u,v}(u,v) &= B_0^0(u)\Delta^{1,1}\mathbf{b}_{0,0}B_0^0(v) \\ &= \Delta^{1,1}\mathbf{b}_{0,0} \\ &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{aligned}$$

$B_0^0(u) = 1$ for all u
$\Rightarrow$ a bilinear patch has a *constant* twist vector

Higher Order Derivatives

The Bernstein basis functions property:

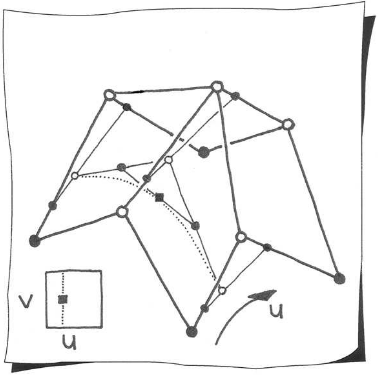
$$B_0^n(0) = 1 \quad \text{and} \quad B_i^n(0) = 0 \quad \text{for } i = 1, n$$
$$B_n^n(1) = 1 \quad \text{and} \quad B_i^n(1) = 0 \quad \text{for } i = 0, n-1$$

$\Rightarrow$ Simple form of the twist at the corners of the patch

$$\mathbf{x}_{u,v}(0,0) = mn\Delta^{1,1}\mathbf{b}_{0,0} \qquad \mathbf{x}_{u,v}(0,1) = mn\Delta^{1,1}\mathbf{b}_{0,n-1}$$
$$\mathbf{x}_{u,v}(1,0) = mn\Delta^{1,1}\mathbf{b}_{m-1,0} \qquad \mathbf{x}_{u,v}(1,1) = mn\Delta^{1,1}\mathbf{b}_{m-1,n-1}$$

The de Casteljau Algorithm

Evaluation of a Bézier patch: $\mathbf{x}(u, v) = M^T \mathbf{B} N$



Define an intermediate set of points

$$\mathbf{C} = M^{\mathrm{T}} \mathbf{B}$$

$$\begin{aligned}
\mathbf{c}_0 &= B_0^m(u)\mathbf{b}_{0,0} + \ldots + B_m^m(u)\mathbf{b}_{m,0} \\
\mathbf{c}_1 &= B_0^m(u)\mathbf{b}_{0,1} + \ldots + B_m^m(u)\mathbf{b}_{m,1} \\
&\ldots \\
\mathbf{c}_n &= B_0^m(u)\mathbf{b}_{0,n} + \ldots + B_m^m(u)\mathbf{b}_{m,n}
\end{aligned}$$

Evaluate n degree m curves with the de Casteljau algorithm

The de Casteljau Algorithm

Final evaluation step: $\mathbf{x}(u, v) = \mathbf{C}N$
$\Rightarrow$ Evaluate this degree n Bézier curve with the de Casteljau algorithm

The 2-stage de Casteljau evaluation method
– Repeated calls to the de Casteljau algorithm for curves

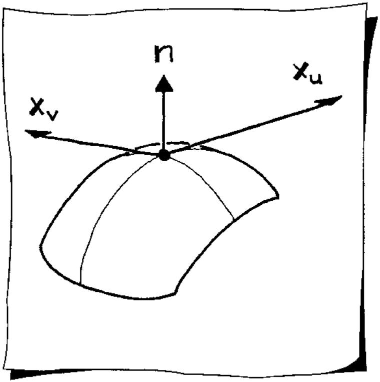
Advantage of this geometric approach:
– Allows computation of a point and derivative

Control polygon $\mathbf{C}$:
– Evaluate point $\mathbf{x}(u, v) = \mathbf{C}N$ and tangent $\mathbf{x}_v$

Control polygon $\mathbf{D} = \mathbf{B}N$:
– Evaluate point $\mathbf{x} = M^{\mathrm{T}}\mathbf{D}$ and tangent $\mathbf{x}_u$

Normals

The normal vector or normal is a fundamental geometric concept
– Used throughout computer graphics and CAD/CAM



At a given point $\mathbf{x}(u, v)$ the normal is *perpendicular* to the surface

Tangent plane at $\mathbf{x}(u, v)$
– Defined by $\mathbf{x}, \mathbf{x}_u, \mathbf{x}_v$
$\Rightarrow$ A point and two vectors

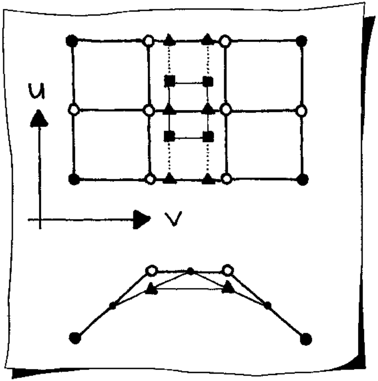
The normal $\mathbf{n}$ is a unit vector defined by

$$\mathbf{n} = \frac{\mathbf{x}_u \wedge \mathbf{x}_v}{\|\mathbf{x}_u \wedge \mathbf{x}_v\|}$$

Normals

3-stage de Casteljau evaluation method

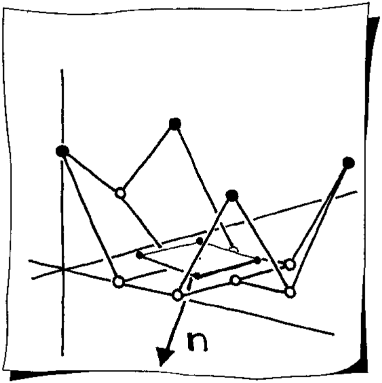
Ingredients for $\mathbf{n}$ are $\mathbf{x}, \mathbf{x}_u$, and $\mathbf{x}_v$



1. For all $m+1$ rows
   Compute $n-1$ levels of dCA
   – v parameter $\rightarrow$ triangles
2. Compute $m-1$ levels of dCA
   – parameter $u \rightarrow$ squares
3. Four points (squares) form a bilinear patch
   – Its tangent plane is surface's tangent plane
   – Evaluate and compute the partials
   – Vectors must be scaled for original patch

Normals

Example: 3-stage de Casteljau evaluation method at $(u, v) = (0.5, 0.5)$



Results in a bilinear patch

Bilinear patch shares the same tangent plane as the original patch $\mathbf{x}$

$$\mathbf{n} = \begin{bmatrix} -0.1240 \\ 0 \\ -0.9922 \end{bmatrix}$$

Changing Degrees

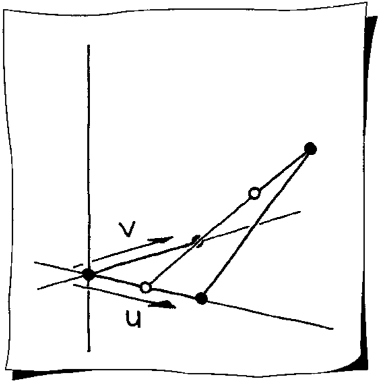
Bézier patch degrees: m in $u-$direction and n in $v-$direction

Degree elevation for curves used to degree elevate patch

Example: Raise m to $m+1$ then resulting control net will have
– $n+1$ columns of control points
– Each column contains $m+2$ control points
– Still describes same surface

Degree reduction performed on a row-by-row or column-by-column basis
– Repeatedly applying the curve algorithm

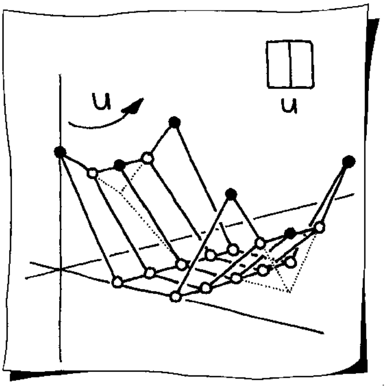
Changing Degrees



Degree elevation of a bilinear patch
– Elevate to degree 2 in u

Subdivision

Curve subdivision: Splitting one curve segment into two segments

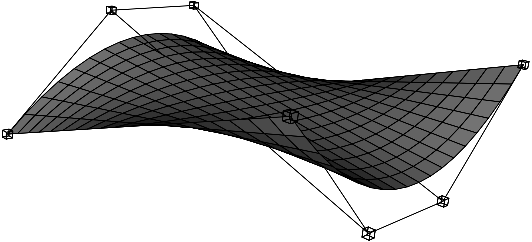


Patch subdivision: split into two patches

Example:
u_0 splits the domain unit square into two rectangles
Patch split along an isoparametric curve

Method:
Perform curve subdivision
for each degree m column of the control net at parameter u_0

Ruled Bézier Patches



Ruled surface is linear in *one* isoparametric direction

$$v\text{-direction linear:} \quad \mathbf{x}(u,v) = (1-v)\mathbf{x}(u,0) + v\mathbf{x}(u,1)$$

$$u\text{-direction linear:} \quad \mathbf{x}(u,v) = (1-u)\mathbf{x}(0,v) + u\mathbf{x}(1,v)$$

⇒ Simple method to fit a surface between two curves
–Two curves the same degree

Example: A bilinear surface

Ruled Bézier Patches

Let the two curves be given

$$u = 0 : \quad \mathbf{b}_{0,0}, \ldots, \mathbf{b}_{m,0} \quad \text{and} \quad u = 1 : \quad \mathbf{b}_{0,1}, \ldots, \mathbf{b}_{m,1}$$

Ruled surface:

$$\mathbf{x}(u,v) = [B_0^m(u), \ldots, B_m^m(u)] \begin{bmatrix} \mathbf{b}_{0,0} & \mathbf{b}_{0,1} \\ \vdots & \vdots \\ \mathbf{b}_{m,0} & \mathbf{b}_{m,1} \end{bmatrix} \begin{bmatrix} B_0^1(v) \\ B_1^1(v) \end{bmatrix}$$

A developable surface is a special ruled surface
– Important in manufacturing
– Bending a piece of sheet metal without tearing or stretching
– Special conditions for a ruled surface to be developable
(Gaussian curvature must be zero everywhere)

Functional Bézier Patches

Functional or nonparametric Bézier patches are analogous to their curve counterparts

The graph of a functional surface is a parametric surface of the form:

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

Important feature: single-valued

$\Rightarrow$ Useful in some applications such as sheet metal stamping

Functional Bézier Patches

Control points for a functional Bézier patch defined over $[0,1] \times [0,1]$

$$\mathbf{b}_{i,j} = \begin{bmatrix} i/m \\ j/n \\ b_{i,j} \end{bmatrix}$$

Over an arbitrary rectangular region $[a,b] \times [c,d]$:

(Direct generalization of functional Bézier curves over an arbitrary interval)

Monomial Patches

$$\mathbf{x}(u,v) = \begin{bmatrix} 1 & u \ldots & u^m \end{bmatrix} \begin{bmatrix} \mathbf{a}_{0,0} & \ldots & \mathbf{a}_{0,n} \\ \vdots & & \vdots \\ \mathbf{a}_{m,0} & \ldots & \mathbf{a}_{m,n} \end{bmatrix} \begin{bmatrix} 1 \\ v \\ \vdots \\ v^n \end{bmatrix}$$

$$= U^{\mathrm{T}} \mathbf{A} V$$

Analogous to curves:

– $\mathbf{a}_{0,0}$ represents a point on the patch at $(u,v) = (0,0)$

– All other $\mathbf{a}_{i,j}$ are partial derivatives

Conversion between monomial and the Bézier forms:

– Analogous to curves

$$\mathbf{a}_{i,j} = \binom{m}{i}\binom{n}{j}\Delta^{i,j}\mathbf{b}_{0,0}$$