

02561 Computer Graphics

Spline Surfaces and Subdivision

Jeppe Revall Frisvad

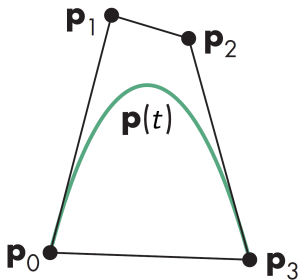
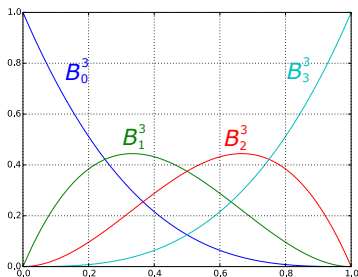
November 2021

Bézier curves

- ▶ A parametric curve is a vector valued function of a single variable.
- ▶ Bernstein basis functions: $B_k^n(t) = \binom{n}{k} (1-t)^{n-k} t^k$, $\binom{n}{k} = \frac{n!}{k!(n-k)!}$.
- ▶ Bézier curve of degree n :

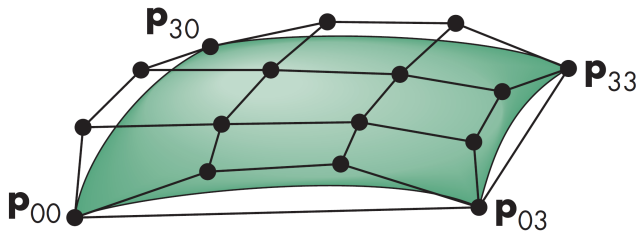
$$\mathbf{p}(t) = \mathbf{P} \begin{bmatrix} B_0^n(t) & B_1^n(t) & \cdots & B_n^n(t) \end{bmatrix}^T, \quad \mathbf{P} = \begin{bmatrix} \mathbf{p}_0 & \mathbf{p}_1 & \cdots & \mathbf{p}_n \end{bmatrix},$$

where \mathbf{p}_k are $n + 1$ Bézier control points.



Tensor product Bézier surface

- ▶ Bézier curve: $\mathbf{p}(t) = \sum_{k=0}^n \mathbf{p}_k B_k^n(t)$.
- ▶ Bézier surface: $\mathbf{p}(u, v) = \sum_{i=0}^n \sum_{j=0}^m \mathbf{p}_{ij} B_i^n(u) B_j^m(v)$.
- ▶ The surface is a tensor product of two curves.



Finding a point on a curve or on a surface

- ▶ Let us define operators producing a modified set of control points:

$$R : (\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_k) \mapsto (\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{k-1})$$

$$L : (\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_k) \mapsto (\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_k).$$

- ▶ We can then define the parameter-dependent *de Casteljau* operator

$$C(t) = (1 - t)R + tL.$$

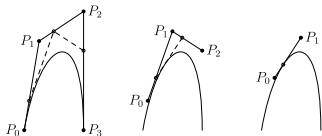
- ▶ A point on a Bézier curve is then given by n -fold application of $C(t)$:

$$\mathbf{p}(t) = C(t)^n \mathbf{P}.$$

- ▶ A point on a Bézier surface is then given by

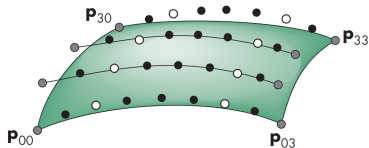
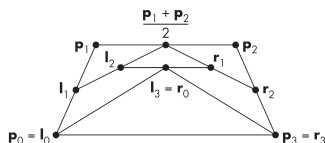
$$\mathbf{p}(u, v) = C_1(u)^n C_2(v)^m \mathbf{P},$$

where \mathbf{P} is a control point mesh.



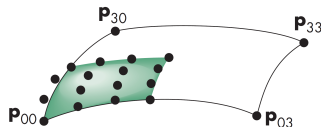
Subdivision

- ▶ The points generated by $C(\frac{1}{2})^n$ can be used as a refined control point mesh.



- New points created by subdivision
- Old points discarded after subdivision
- Old points retained after subdivision

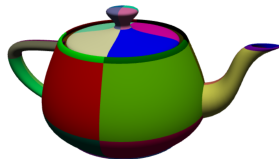
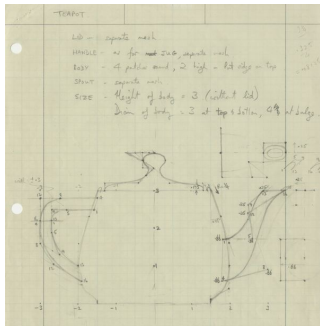
- ▶ In the case of a surface, we apply the de Casteljau operator to each curve in the tensor product.
- ▶ The result is four connected quadrants with the same number of control points as the original control point mesh.
- ▶ The control mesh generated by recursive subdivision converges to the actual Bézier curve or Bézier surface that it describes.
- ▶ We can thus use the control mesh as a subdivision surface.



Newell's teapot

- ▶ One of the first computer graphics objects to be modeled using spline surfaces.

Computer History Museum. Michael Hicks from Saint Paul, MN, USA. CC BY 2.0, via Wikimedia Commons



- ▶ This Utah teapot consists of 32 bicubic Bézier patches defined by 302 vertices.
- ▶ The textbook describes an implementation of the teapot as a subdivision surface.
- ▶ Some patches are degenerate, which means that care must be taken when computing surface normals.

Differentiation to get surface normals

- ▶ The *forward difference* operator: $\Delta = L - R$.
- ▶ If applied to a Bézier curve of degree n , the result is a Bézier curve of degree $n - 1$.
- ▶ We observe that $C(t) = (1 - t)R + tL = R + t\Delta$, and we have $\frac{d}{dt}C(t) = \Delta$.
- ▶ For surfaces, we use a subscript to denote whether the operator pertains to rows (Δ_1) or columns (Δ_2) in the control mesh \mathbf{P} .
- ▶ We then have the following differential vectors:

$$\begin{aligned}\frac{\partial \mathbf{p}}{\partial u}(u, v) &= n C_1(u)^{n-1} C_2(v)^m(v) \Delta_1 \mathbf{P} \\ \frac{\partial \mathbf{p}}{\partial v}(u, v) &= m C_1(u)^n C_2(v)^{m-1}(v) \Delta_2 \mathbf{P}\end{aligned}$$

and the surface normal is $\mathbf{n}(u, v) = \frac{\partial \mathbf{p}}{\partial u}(u, v) \times \frac{\partial \mathbf{p}}{\partial v}(u, v)$.

- ▶ We can use this to get vertex normals for a given subdivision of the control mesh.

WebGL demonstrator (**A**: 11.10)

subdivide

coarsen

