Splines

Spline Curves

- Successive linear blend
- Basis polynomials
- Recursive evaluation
- Properties
- Joining segments

Tensor-product-patch Spline Surfaces

- Tensor product patches
- Evaluation
- Properties
- Joining patches

Triangular-patch Spline Surfaces

- Coordinate frames and barycentric frames
- Triangular patches
Discontinuities

- Basis polynomials
- Multiple segments
- Basis splines

Continuities

- Combining basis splines for smoothness
- Curves with basis splines

B-Splines

- General segmentation and smoothness
- Knots and evaluation
Constructing Curve Segments

Linear blend:

- Line segment from an affine combination of points

\[ P_0^1(t) = (1 - t)P_0 + tP_1 \]
**Quadratic blend:**

- Quadratic segment from an affine combination of line segments

\[
P_0^1(t) = (1 - t)P_0 + tP_1
\]
\[
P_1^1(t) = (1 - t)P_1 + tP_2
\]
\[
P_0^2(t) = (1 - t)P_0^1(t) + tP_1^1(t)
\]
Cubic blend:

- Cubic segment from an affine combination of quadratic segments

\[
\begin{align*}
P_0^1(t) & = (1 - t)P_0 + tP_1 \\
P_1^1(t) & = (1 - t)P_1 + tP_2 \\
P_0^2(t) & = (1 - t)P_0^1(t) + tP_1^1(t) \\
P_1^1(t) & = (1 - t)P_1 + tP_2 \\
P_2^1(t) & = (1 - t)P_2 + tP_3 \\
P_1^2(t) & = (1 - t)P_1^1(t) + tP_2^1(t) \\
P_0^3(t) & = (1 - t)P_0^2(t) + tP_1^2(t)
\end{align*}
\]

- The pattern should be evident for higher degrees
Geometric view (deCasteljau Algorithm):

- Join the points $P_i$ by line segments
- Join the $t : (1 - t)$ points of those line segments by line segments
- Repeat as necessary
- The $t : (1 - t)$ point on the final line segment is a point on the curve
- The final line segment is tangent to the curve at $t$
Expanding Terms (Basis Polynomials):

- The original points appear as coefficients of Bernstein polynomials

\[
P_0^0(t) = P_0 \quad 1
\]
\[
P_0^1(t) = (1 - t)P_0 + tP_1
\]
\[
P_0^2(t) = (1 - t)^2P_0 + 2(1 - t)tP_1 + t^2P_2
\]
\[
P_0^3(t) = (1 - t)^3P_0 + 3(1 - t)^2tP_1 + 3(1 - t)t^2P_2 + t^3P_3
\]
\[
P_0^n(t) = \sum_{i=0}^{n} P_i B_i^n(t)
\]

where

\[
B_i^n(t) = \frac{n!}{(n - i)!i!} (1 - t)^{n-i} t^i = \binom{n}{i} (1 - t)^{n-i} t^i
\]

- The Bernstein polynomials of degree \(n\) form a basis for the space of all degree-\(n\) polynomials
Recursive evaluation schemes:

- To obtain curve points:
  - Start with given points and form successive, pairwise, affine combinations

  \[ P_i^0 = P_i \]
  \[ P_i^j = (1 - t)P_i^{j-1} + tP_{i+1}^{j-1} \]

  - The generated points \( P_i^j \) are the deCasteljau points

- To obtain basis polynomials:
  - Start with 1 and form successive, pairwise, affine combinations

  \[ B_0^0 = 1 \]
  \[ B_i^j = (1 - t)B_i^{j-1} + tB_{i+1}^{j-1} \]

  where \( B_r^s = 0 \) when \( r < 0 \) or \( r > s \)
Recursive triangle diagrams (upward):

Computing deCasteljau points

- Each node gets the affine combination of the two nodes entering from below
  - Leaf nodes have the value of their respective points
    \[ P_1^2 = (1 - t)P_1^1 + tP_2^1 \]

- Each node gets the sum of the path products entering from below
  \[ P_1^2 = P_0^1(1 - t)(1 - t) + P_0^2 t(1 - t) + P_0^2 (1 - t)t + P_3^0 tt \]
  \[ P_1^2 = (1 - t)^2 P_0^1 + 2(1 - t)t P_0^2 + t^2 P_3^0 \]
\[
\begin{align*}
P_0^3 & \quad (1-t) \quad t \\
      & \quad P_0^2 \\
      & \quad (1-t) \quad t \quad (1-t) \\
      & \quad P_0^1 \\
      & \quad (1-t) \quad t \quad (1-t) \\
      & \quad P_0^0 \\
      & \quad P_1^0 \\
      & \quad P_1^1 \\
      & \quad P_2^1 \\
      & \quad P_3^0 \\
P_0 & \quad P_1 & \quad P_2 & \quad P_3
\end{align*}
\]
Recursive triangle diagrams (downward):

Computing Bernstein (basis) polynomials

- Each node gets the affine combination of the two nodes entering from above
  - Root node has value 1
  - For other nodes, missing entries above count as zero
- Each node gets the sum of the path products entering from above

\[
B_1^3 = t(1 - t)(1 - t) + (1 - t)t(1 - t) + (1 - t)t(1 - t)t
\]

\[
P_1^3 = 3(1 - t)^2t
\]
\[
\begin{align*}
B_0^0 & \quad (1-t) \quad t \\
B_0^1 & \quad B_1^1 \\
B_0^2 & \quad B_1^2 \\
B_0^3 & \quad B_1^3 \\
B_0^3 & \quad B_1^3 \\
B_0^3 & \quad B_1^3 \\
B_0^3 & \quad B_1^3 \\
B_0^3 & \quad B_1^3 \\
\end{align*}
\]
Bernstein Basis Functions

Bernstein Polynomial Properties:

Partition of Unity: \( \sum_{i=0}^{n} B_i^n(t) = 1 \)

Proof:

\[
1 = (t + (1 - t))^n \\
= \sum_{i=0}^{n} \binom{n}{i} (1 - t)^{n-i} t^i \\
= \sum_{i=0}^{n} B_i^n(t)
\]

Nonnegativity: \( B_i^n(t) \geq 0 \), for \( t \in [0, 1] \)

Proof:

\[
\binom{n}{i} > 0 \\
\quad t \geq 0 \text{ for } 0 \leq t \leq 1 \\
(1 - t) \geq 0 \text{ for } 0 \leq t \leq 1
\]
Recurrence: $B_0^0(t) = 1$ and $B_i^n(t) = (1 - t)B_i^{n-1}(t) + B_{i-1}^{n-1}(t)$

Proof:

\[
B_i^n(t) = \binom{n}{i} t^i (1 - t)^{n-i} \\
= \binom{n-1}{i} t^i (1 - t)^{n-i} + \binom{n-1}{i-1} t^i (1 - t)^{n-i} \\
= (1 - t) \binom{n-1}{i} t^i (1 - t)^{(n-1)-i} + \\
t \binom{n-1}{i-1} t^{i-1} (1 - t)^{(n-1)-(i-1)} \\
= (1 - t)B_i^{n-1}(t) + tB_{i-1}^{n-1}(t)
\]
Derivatives: \( \frac{d}{dt} B_i^n(t) = n \left( B_i^{n-1}(t) - B_i^{n-1}(t) \right) \)

Proof:

\[
\frac{d}{dt} B_i^n(t) = \frac{d}{dt} \left( \binom{n}{i} t^i (1 - t)^{n-i} \right) \\
= \frac{d}{dt} \left( \frac{i!(n - i)!}{n!} t^i (1 - t)^{n-i} \right) \\
= \frac{i!(n - i)!}{in!} t^{i-1} (1 - t)^{n-i-1} \\
= \frac{(i - 1)!(n - i)!}{n(n - 1)!} t^{i-1} (1 - t)^{n-i} - \\
\frac{i!(n - i - 1)!}{n(n - i)!} t^i (1 - t)^{n-i-1} \\
= n \left( B_i^{n-1}(t) - B_i^{n-1}(t) \right)
\]
Bézier Splines

Bézier Curve Segments and their Properties

Definition:

- A degree $n$ (order $n + 1$) Bézier curve segment is

$$P(t) = \sum_{i=0}^{n} P_i B_i^n(t)$$

where the $P_i$ are $k$-dimensional control points.

Convex Hull:

$$\sum_{i=0}^{n} B_i^n(t) = 1, \quad B_i^n(t) \geq 0 \text{ for } t \in [0, 1]$$

$\implies$ $P(t)$ is a convex combination of the $P_i$ for $t \in [0, 1]$

$\implies$ $P(t)$ lies within convex hull of $P_i$ for $t \in [0, 1]$
Affine Invariance:

- A Bézier curve is an affine combination of its control points
- Any affine transformation of a curve is the curve of the transformed control points

\[ T \left( \sum_{i=0}^{n} P_i B_i^n(t) \right) = \sum_{i=0}^{n} T(P_i) B_i^n(t) \]

- *This property does not hold for projective transformations!*

Interpolation:

\[ B_0^n(0) = 1, \; B^n(1) = 1, \; \sum i = 0^n B_i^n(t) = 1, \; B_i^n(t) \geq 0 \]

for \( t \in [0, 1] \)

\[ \implies B_i^n(0) = 0 \text{ if } i \neq 0, \; B_i^n(1) = 0 \text{ if } i \neq n \]

\[ \implies P(0) = P_0, \; P(1) = P_n \]
Derivatives:

\[
\frac{d}{dt} B^n_i(t) = n \left( B^n_{i-1}(t) - B^n_{i-1}(t) \right)
\]

\[\implies P'(0) = n(P_1 - P_0), \quad P'(1) = n(P_n - P_{n-1})\]

Smoothly Joined Segments ($G^1$):

- Let $P_{n-1}, P_n$ be the last two control points of one segment
- Let $Q_0, Q_1$ be the first two control points of the next segment

\[
P_n = Q_0
\]

\[
(P_n - P_{n-1}) = \beta (Q_1 - Q_0) \text{ for some } \beta > 0
\]
Recurrence, Subdivision:

\[ B_i^n(t) = (1 - t)B_i^{n-1} + tB_{i-1}^{n-1}(t) \]

\[ \implies \text{deCasteljau’s algorithm:} \]

\[ P(t) = P_o^n(t) \]

\[ P_i^k(t) = (1 - t)P_i^{k-1}(t) + tP_{i+1}^{k-1} \]

\[ P_i^0 = P_i \]

Use to evaluate point at \( t \), or subdivide into two new curves:

- \( P_0^0, P_0^1, \ldots P_0^n \) are the control points for the left half
- \( P_n^0, P_{n-1}^1, \ldots P_0^n \) are the control points for the right half
Matrix View:

- Expand each Bernstein polynomial in powers of $t$
- Represent each expansion as the column of a matrix
- Quadratic example:

$$
(1 - t)^2 P_0 + 2(1 - t)t P_1 + t^2 P_2
= (1 - 2t + t^2) P_0 + (2t - 2t^2) P_0 + (2t - 2t^2) P_1 + t^2 P_2
= \begin{bmatrix} 1 & t & t^2 \end{bmatrix} \begin{bmatrix}
1 & 0 & 0 \\
-2 & 2 & 0 \\
1 & -2 & 1 
\end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \end{bmatrix}
$$

- In matrix format:

$$P(t) = T(t)^T M_{BT} P$$

- $T(t)^T = [1 \ t \ t^2]$ is the monomial basis
- $P_T = M_{BT} P$ is a matrix containing the coefficients of the polynomials for each dimension of $P(t)$
- $M_{BT}$ is a change of basis matrix that converts a specification $P$ of $P(t)$ relative to the Bernstein basis to one relative to the monomial basis
Tensor Product Patches

Tensor Product Patches:

- The control polygon is the polygonal mesh with vertices \( P_{i,j} \)
- The patch basis functions are products of curve basis functions

\[
P(s, t) = \sum_{i=0}^{n} \sum_{j=0}^{n} P_{i,j} B_{i,j}^{n}(s, t)
\]

where

\[
B_{i,j}^{n}(s, t) = B_{i}^{n}(s) B_{j}^{n}(t)
\]

Scan in image.
Properties:

- Patch basis functions *sume to one*

\[
\sum_{i=0}^{n} \sum_{j=0}^{n} B_i^n(s)B_j^n(t) = 1
\]

- Patch basis functions are *nonnegative* on \([0, 1] \times [0, 1]\\)

\[
B_i^n(s)B_j^n(t) \geq 0 \text{ for } 0 \leq s, t \leq 1
\]

\[\Rightarrow \text{ Surface patch is in the convex hull of the control points}\\
\Rightarrow \text{ Surface patch is affinely invariant}\\
\text{(Transform the patch by transforming the control points)}\\
\]

*Subdivision, Recursion, Evaluation:*

- As for curves in each variable separately and independently
- *Tangent plane is not produced!*
  - Normals must be computed from partial derivatives
Partial Derivatives:

- Ordinary derivative in each variable separately:

\[
\frac{\partial}{\partial s} P(s, t) = \sum_{i=0}^{n} \sum_{j=0}^{n} P_{i,j} \left[ \frac{d}{ds} B^n_i(s) \right] B^n_j(t)
\]

\[
\frac{\partial}{\partial s} P(s, t) = \sum_{i=0}^{n} \sum_{j=0}^{n} P_{i,j} B^n_i(s) \left[ \frac{d}{dt} B^n_j(t) \right]
\]

- Each of the above is a tangent vector in a parametric direction
- Surface is regular at each \((s, t)\) where these two vectors are linearly independent
- The (unnormalized) surface normal is given at any regular point by

\[
\pm \left[ \frac{\partial}{\partial s} P(s, t) \times \frac{\partial}{\partial t} P(s, t) \right]
\]

(the sign dictates what is the outward pointing normal)

- In particular, the cross-boundary tangent is given by (e.g., for the \(s = 0\) boundary):

\[
n \sum_{i=0}^{n} \sum_{j=0}^{n} (P_{1,j} - P_{0,j}) B^n_j(t)
\]

(and similarly for the other boundaries)
Smoothly Joined Patches:

- Can be achieved by ensuring that

\[(P_{i,n} - P_{i,n-1}) = \beta(Q_{i,1} - Q_i, 0) \text{ for } \beta > 0\]

(and correspondingly for other boundaries)
Rendering:

- Divide up into polygons:
  1. By stepping

\[
\begin{align*}
  s &= 0, \delta, 2\delta, \ldots, 1 \\
  t &= 1, \gamma, 2\gamma, \ldots, 1
\end{align*}
\]

and joining up sides and diagonals to produce a triangular mesh

2. By subdividing and rendering the control polygon
Barycentric Coordinates (optional)

Coordinate Frames:

- Vector oriented; derived from linear space basis
- One point and \( n \) vectors in space of dimension: \( n : D_n, \vec{v}_0, \ldots , \vec{v}_{n-1} \)
  - Vectors \( \vec{v}_i \) are linearly independent

Barycentric Frames:

- Point oriented
- \( n + 1 \) points in space of dimension \( N : D_0, \ldots , D_n \)
  - Points are in general position
Frames of Both Types Are Equivalent

- Express each $\vec{v}_i$ as $D_i - D_n$ for $D_i = D_n + v_i$

$$ P = D_n + \sum_{i=0}^{n-1} p_i \vec{v}_i $$

$$ = D_n + \sum_{i=0}^{n-1} p_i (D_i - D_n) $$

$$ = (1 - \sum_{i=0}^{n-1} p_i) D_n + \sum_{i=0}^{n-1} p_i D_i $$

$$ = \sum_{i=0}^{n-1} w_i D_i \text{ where } \sum_{i=0}^{n-1} w_i = 1 $$

- And, of course, conversely
**Triangular Patches (optional)**

*deCasteljau Revisited Barycentrically:*

- Linear blend expressed in barycentric terms

\[
(1 - t)P_0 + tP_1 = rP_0 + tP_1 \text{ where } r + t = 1
\]

- Higher powers and a symmetric form of the Bernstein polynomials:

\[
P(t) = \sum_{i=0}^{n} P_i \left( \frac{n!}{i!(n-i)!} \right) (1 - t)^{n-i} t^i
\]

\[
= \sum_{\begin{array}{c} i + j = n \\
 i \geq 0, j \geq 0 \end{array}} P_i \left( \frac{n!}{i!j!} \right) t^i r^j \text{ where } r + t = 1
\]

\[
\implies \sum_{\begin{array}{c} i + j = n \\
 i \geq 0, j \geq 0 \end{array}} P_{ij} B_{ij}^n(r, t)
\]
• Examples

\begin{align*}
B_{00}^0(r, t) &= 1 \\
B_{01}^1(r, t), B_{10}^1(r, t) &= r, t \\
B_{02}^2(r, t), B_{11}^1(r, t), B_{20}^2(r, t) &= r^2, 2rt, t^2 \\
B_{03}^3(r, t), B_{12}^3(r, t), B_{21}^3(r, t), B_{30}^3(r, t) &= r^3, 3r^2t, 3rt^2, t^3
\end{align*}

Surfaces – Barycentric Blends on Triangles:

• Formulas

\begin{align*}
P(r, s, t) &= \sum_{i + j + k = n} P_{ijk} B_{ijk}^n(r, s, t) \\
&\quad \text{where } i \geq 0, j \geq 0, k \geq 0 \\
B_{ijk}^n(r, s, t) &= \frac{n!}{i! j! k!} r^i s^j t^k
\end{align*}
Triangular deCasteljau:

- Join adjacently indexed $P_{ijk}$ by triangles
- Find $r : s : t$ barycentric point in each triangle
- Join adjacent points by triangles
- Repeat
  - Final point is the surface point $P(r, s, t)$
  - final triangle is tangent to the surface at $P(r, s, t)$
- Triangle up/down schemes become tetrahedral up/down schemes

Scan in image.
Properties:

- Each boundary curve is a Bézier curve
- Patches will be joined smoothly if pairs of boundary triangles are planar as shown
Discontinuities in Splines

Bézier Discontinuities:

- Two Bézier segments can be completely disjoint
- Two segments join if they share last/first control point
Common Parameterization and Blending Functions

- Joined curves can be given common parameterization
  - Parameterize first segment with $0 \leq t < 1$
  - Parameterize nest segment with $1 \leq t \leq 2$, etc.
- Look at blending/basis polynomials under this parameterization
  - Combine those for common $P_j$ into a single piecewise polynomial

![Diagram of B-spline basis functions]

$B_0 \quad B_1 \quad B_2 \quad B_3 \quad B_4$

0 \quad 1 \quad 2
**Combined Curve Segments**

- Curve is \( P(t) = P_0 B_0(t) + P_1 B_1(t) + P_2 B_2(t) + P_3 B_3(t) + P_4 B_4(t) \), where

\[
\begin{align*}
B_0(t) &= \begin{cases} 
(1 - t)^2 & 0 \leq t < 1 \\
0 & 1 \leq t \leq 2
\end{cases} \\
B_1(t) &= \begin{cases} 
2((1 - t)t) & 0 \leq t < 1 \\
0 & 1 \leq t \leq 2
\end{cases} \\
B_2(t) &= \begin{cases} 
t^2 & 0 \leq t < 1 \\
(2 - t)^2 & 1 \leq t \leq 2
\end{cases} \\
B_3(t) &= \begin{cases} 
0 & 0 \leq t < 1 \\
2(2 - t)(t - 1) & 1 \leq t \leq 2
\end{cases} \\
B_4(t) &= \begin{cases} 
0 & 0 \leq t < 1 \\
(t - 1)^2 & 1 \leq t \leq 2
\end{cases}
\]
Curve Discontinuities from Basis Discontinuities

• $P_2$ is scaled by $B_2(t)$, which has a discontinuous derivative
• The corner in the curve results from this discontinuity
Spline Continuity

**Smother Blending Functions:**

- Can \( B_0(t), \ldots, B_4(t) \) be replaced by smoother functions?
  - Piecewise polynomials on \( 0 \leq t \leq 2 \)
  - Continuous derivatives
- Yes, but we lose one degree of freedom
  - Curve has no corner if segments share a common tangent
  - Tangent is given by the chords \( P_1P_2, P_2P_3 \)
  - An equation constrains \( P_1, P_2, P_3 \)
    \[
    P_3 - P_2 = P_2 - P_1 \implies P_2 = \frac{P_1 + P_3}{2}
    \]
- This equation leads to combinations:

\[
P_0 B_0(t) + P_1 \left( B_1(t) + \frac{1}{2} B_2(t) \right) + P_3 \left( \frac{1}{2} B_2(t) + B_3(t) \right) + P_4 B_4(t)
\]
**Spline Basis:**

- Combined functions form a smoother *spline basis*

\[
\begin{align*}
\overline{B}_0(t) &= B_0(t) \\
\overline{B}_1(t) &= \left( B_1(t) + \frac{1}{2}B_2(t) \right) \\
\overline{B}_2(t) &= \left( \frac{1}{2}B_2(t) + B_3(t) \right) \\
\overline{B}_3(t) &= B_4(t)
\end{align*}
\]
Smother Curves:

- Control points used with this basis produce smoother curves.
B-Splines

General B-Splines:

- Nonuniform B-splines (NUBS) generalize this construction
- A B-spline, $B_i^d(t)$, is a piecewise polynomial:
  - each of its segments is of degree $\leq d$
  - it is defined for all $t$
  - its segmentation is given by knots $t = t_0 \leq t_1 \leq \cdots \leq t_N$
  - it is zero for $T < T_i$ and $T > T_{i+d+1}$
  - it may have a discontinuity in its $d - k + 1$ derivative at $t_j \in \{t_i, \ldots, t_{i+d+1}\}$, if $t_j$ has multiplicity $k$
  - it is nonnegative for $t_i < t < t_{i+d+1}$
  - $B_i^d(t) + \cdots + B_{i+d}(t) = 1$ for $t_i+d \leq t < t_{i+d+1}$, and all other $B_j^d(t)$ are zero on this interval
  - Bézier blending functions are the special case where all knots have multiplicity $d + 1$
Example (Quadratic):
Evaluation:

- There is an efficient, recursive evaluation scheme for any curve point.
- It generalizes the triangle scheme (deCasteljau) for Bézier curves.
- Example (for cubics and $t_{i+3} \leq t < t_{i+4}$):

\[
\begin{align*}
P_i^0 & \quad P_{i+1}^0 \\
P_i & \quad P_{i+1} \\
P_{i+1} & \quad P_{i+1} \\
P_{i+1} & \quad P_{i+2} \\
P_{i+2} & \quad P_{i+3} \\
P_{i+3} & \quad P_{i+3}
\end{align*}
\]