Modelling Curves 2 - Bézier Splines

Bézier Splines

- Bézier Curves were discovered by Pierre Bézier.
- Approximating the shape of a control polygon.
- Mathematically: \( P(t) = \sum_{i=0}^{n} B_i(t) \) with \( 0 \leq t \leq 1 \)
- Where \( J_n,i(t) = \binom{n}{i} t^i (1-t)^{n-i} \) and is called the Bernstein basis.
- \( \binom{n}{i} = \frac{n!}{i!(n-i)!} \)
- Control polygon: \( P(t) \)
- Curve: \( P(t) \)

Bézier Splines

- Cubic Bézier Splines
  \( P(t) = \begin{bmatrix} 1-t \\ 3t^2 \end{bmatrix} \begin{bmatrix} 1 \\ 6 \\ 0 \end{bmatrix} \) with \( 0 \leq t \leq 1 \)
  \( J_3,0(t) = (1-t)^3 \)
  \( J_3,1(t) = 3t(1-t)^2 \)
  \( J_3,2(t) = 3t^2(1-t) \)
  \( J_3,3(t) = t^3 \)
Parameter Domain Transformation:
- Concident with the first and last point of the control polygon.
- The curve generally follows the shape of the control polygon.
- The number of defining control polygon points.
- Degree of the polynomial defining the Bézier spline is one less than the number of defining control polygon points.
- The curve generally follows the shape of the control polygon.

Bézier Splines
- Affine Invariance: Applying an affine transform to the curve is equivalent to applying the transformation to the control points.

\[
M = A + B = M \sum x_i \mathbf{J}_i(t) + \sum y_i \mathbf{J}_i(t) = \sum (A \mathbf{J}_i(t)) + \sum (B \mathbf{J}_i(t))
\]

Tangent Vectors:
- So every point lies in the convex hull of the control points.

\[
\mathbf{J}_i(t) = \begin{cases} \mathbf{x} & t \geq 0 \\ \mathbf{y} & t < 0 \end{cases}
\]

Convex Hull: The curve lies inside the convex hull of the control points.
- Given a set of points \(X = \{x_0, x_1, \ldots, x_n\}\) the convex hull of \(X\) is given by the set of points:

\[
C(X) = \{x \in \mathbb{R}^n : \sum_{i=0}^{n} \alpha_i x_i = x \; \text{where} \; \sum_{i=0}^{n} \alpha_i = 1, \alpha_i \geq 0 \; \forall i \}
\]

Every point on the curve is of the form:

\[
P(t) = \sum_{i=0}^{n} \mathbf{J}_i(t) = 0, 1, \ldots, n, \mathbf{J}_i(t) \in \mathbb{R}
\]

So every point lies in the convex hull of \(\mathbf{B}_i\)

Parameter Domain Transformation:
- Symmetry: \(J_i(t) = J_i(1-t)\)
- The curve \(P(t)\) formed by the control points \(B_0, \ldots, B_n\) is the same as the curve \(P(1-t)\) formed by the control points \(B_n-1, \ldots, B_0\).

Parameter Domain Transformation:
- \(t \in [0,1], \alpha \in [a, b]\)

\[
P(t) = \sum_{i=0}^{n} \mathbf{J}_i(t) = \sum_{i=0}^{n} \mathbf{B}_i \cdot \frac{a - \alpha}{b - a}
\]
Bézier Splines

- Tangent Vectors:
  \[ P'(0) = n \overrightarrow{B_1 - B_0} \]
  \[ P'(1) = n \overrightarrow{B_n - B_{n-1}} \]
  i.e., tangent vectors at the ends of curve have the same direction as the first and last spans of the control polygon.

- Continuity: \( P(t) \) of degree \( n \), defined by control vertices \( B_i \)
  \( Q(s) \) of degree \( m \), defined by control vertices \( C_i \)
  \( P'(0) = n \overrightarrow{B_1 - B_0} \implies C_1 - C_0 = n \overrightarrow{B_n - B_{n-1}} \implies C_1 = n \overrightarrow{B_n - B_{n-1}} + B_n \) because \( C_0 = B_n \)
  i.e., \( B_n, C_0, C_1 \) have to be collinear for the tangents to have the same direction.

Bézier Splines

- Control:
  - How does the shape of the curve change if we move one control point?
  - Each control point is associated to a basis function.
  - The basis functions effect the shape of the curve over a range of parameter values where the basis function is non-zero. In case of the Bernstein basis, this is the entire parameter range [0,1].

Bézier Splines

- Points on the curve: De Casteljau’s Algorithm
  deCasteljau(int i, int j)
  {
    if (i == 0) return \( B_{i,j} \);
    else return \( (1-u) \times \text{deCasteljau}(i-1,j) + u \times \text{deCasteljau}(i-1,j+1) \);
  }

  Is this a good way to implement the algorithm?
  Why bother with the algorithm at all?
  Subdivision and Degree Elevation.