Modelling Curves 1 - Cubic Splines

Modelling
- Create the virtual world
  - Create objects
  - Create animals, humans, and aliens too.

Curves
- Curves allow us to design and create complex geometry.

Curves
- Linear Approximation
  - Easy but not good approximation
  - Not smooth

Curves
- Higher Degree Approximations
  - Explicit
    \[ y = f(x) \]
  - Implicit
    \[ f(x, y) = 0 \]
  - Parametric
    \[ x = f_1(t), y = f_2(t) \]

Curves
- Explicit Representation
  - Function plot over some interval
  - Simple to compute and plot
  - Simple to check whether a point lies on the curve.
  - Problem with closed and multi-valued curves
Curves

- Implicit Representation
  - \( f(x, y) = 0 \)
  - The dependent variable is not given 'explicitly' in terms of the independent variable
  - Curves defined implicitly as solution of a system of equations.
  - For e.g., \( Ax + Bx + C = 0, x^2 + y^2 - R^2 = 0 \)
- Harder to render.
- Simple to check whether a point lies on the curve.
- Can represent closed and multi-valued curves.

Cubic Splines

\[
P'(x) = B_1 x + B_2 x^2 + B_3 x^3 = \sum_{i=1}^{4} B_i x^{i-1} \quad \text{with} \quad t_1 \leq x \leq t_2
\]

\[
x(t) = \sum_{i=1}^{4} B_i x^{i-1} \quad \text{with} \quad t_1 \leq x \leq t_2
\]

\[
y(t) = \sum_{i=1}^{4} B_i y^{i-1} \quad \text{with} \quad t_1 \leq x \leq t_2
\]

\[
P''(x) = 2B_2 x + 3B_3 x^2 = \sum_{i=1}^{4} (i-1) B_i x^{i-2} \quad \text{with} \quad t_1 \leq x \leq t_2
\]

- Parametric Representation
  - \( x = \frac{x}{f_x(t)}, y = \frac{y}{f_y(t)} \)
  - Position on the curve is given in terms of a parameter.
  - For e.g., \( x(t) = A(1 - t^2) + Bt, y(t) = A(1 - t^2) + Bt, \) \( t \in [0,1] \)
  - Position depends on \( \cos(t), \) \( \sin(t), \) \( t \in [0,1] \)
- Simple to render.
- Harder to check whether a point lies on the curve.
- Can represent closed and multi-valued curves.

Parametric Curves

- Can represent a variety of curves.
- Can be used for:
  - Interpolation
  - Approximation
- Splines
  - Cubic, Hermite, Bezier, B-Splines, NURBS
  - Specification, Control, Editing
Smoothness?

But what is this notion of smoothness at the join to ensure smoothness?

We will do a piecewise polynomial interpolation with some constraints.

Cubic Splines

Substituting and solving we get:

$$t_1 P_1'' + 2 (t_2 - t_1) P_1' + t_2 P_1'' = \frac{1}{t_2 - t_1} (t_1 (P_2 - P_1) + t_2 (P_2 - P_1))$$

Continuity

Geometric

- $G_1$: Curves are joined
- $G_2$: First derivatives are proportional. Tangents have same directions but not necessarily the same magnitude
- $G_3$: First and second derivatives are proportional across the point of joining.

Parametric continuity of order $n$ implies geometric continuity, but not vice versa.

Cubic Splines

Similarly for $n^2$ and $n + 1$ segments with $1 \leq k \leq m - 2$:

$$[x_{k+1} 2 (t_{k+1} - t_k) x_{k+1} P_{k+1}'' P_{k+1}] = \frac{3}{t_{k+1} - t_k} ((t_{k+1} - t_k) P_{k+1}'' - (t_{k+1} - t_k) P_{k+1}'')$$

If we stack up the equations for all the tangent vectors we get a set of $n-2$ equations.

Cubic Splines

So we enforce $C^1$ continuity at the in-between point.

- For the first segment
  $$P_r''(x) = \frac{1}{t_{r+1} - t_r} (t_{r+1} (P_{r+1} - P_r) + t_r (P_{r+1} - P_r))$$

- For the second segment
  $$P_{r+1}'(x) = 2 B_{r+1} t_{r+1} - 2 B_r t_r$$

$B_0, B_1, B_2, \ldots$ is the Bernstein basis function.

Please note the two $B_k$'s in the equation are from different spans.
Cubic Splines

Now, for a curve segment of the cubic spline,
\[ P_i(t) = \sum_{j=0}^{n} B_j(t) P_j \]

with \( 0 \leq t \leq 1 \) and \( i = 1, \ldots, n-1 \)

\[ B_j(t) = \begin{cases} 1 & \text{if } j = 0, t^3 \leq j^3 \leq (j+1)^3 \leq t^3, \text{ and } 0 \leq j \leq n-1 \end{cases} \]

These are called the blending or weighting functions.

Solving for \( B_k, B_l, B_r, \text{ and } B_s \)

or it can be rewritten as:

\[
\begin{bmatrix}
B_0 \\
B_1 \\
B_2 \\
B_3 \\
\vdots \\
B_n
\end{bmatrix} =
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
P_0 \\
P_1 \\
P_2 \\
P_3 \\
\vdots \\
P_n
\end{bmatrix}
\]

Curve passes through \( P_i \) with endpoint constraints vs tangent constraints.

Influence of endpoints constraints vs tangent constraints.

\( P_i(t) = F \cdot B \)

where \( F \) is the blending function matrix and \( B \) is the geometry matrix.

\( F(0) = 1, F(1) = F(0) = F(1) = 0 \)

\( F(1) = 0, F(1) = 1, F(0) = 0 \)

Curve passes through \( P_i \).

\( F(1) = 1 - P_i(t), F(1) = P_i(1-t) \)

Curve passes through \( P_i \).
Cubic Splines

- Piecewise cubic splines are specified using: position vectors of end points, tangent vectors of end points and parameter value \( t_k \).

- Effect of tangent magnitude


Cubic Splines

- Piecewise cubic splines are specified using: position vectors of end points, tangent vectors of end points and parameter value \( t_k \).

- Effect of parameterization

- Normalized parametrization all \( t_k = 1 \).

- Chord length parametrization

Normalized Cubic Splines: Hermite Splines

The tridiagonal system becomes:

\[
\begin{bmatrix}
F_1(t) & F_2(t) & F_3(t) & F_4(t)
\end{bmatrix}
= \begin{bmatrix}
2 & -2 & 1 & 0 \\
-2 & 4 & -2 & 0 \\
2 & -3 & 2 & -1 \\
0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
P_1' \\
P_2' \\
P_3' \\
P_4'
\end{bmatrix}
\]

Cubic Splines

- Various end conditions for Cubic Splines

  - Clamped: \( P_1'(0) = P_1' \) and \( P_n'(t_n) = P_n' \) are known
  - Relaxed/Natural: \( P_1''(0) = P_n''(t_n) = 0 \)
  - Cyclic: \( P_1'(0) = P_n'(t_n) \) and \( P_1''(0) = P_n''(t_n) \)
  - Anti-cyclic: \( P_1'(0) = -P_n'(t_n) \) and \( P_1''(0) = -P_n''(t_n) \)