Modelling Curves 1 - Cubic Splines
Modelling

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

http://www.its-ming.com/images/bruce_wireframe.jpg

http://www.gametrailers.com/users/druie/gamepad/
Curves

- Curves allow us to design and create complex geometry.

http://www.gametrailers.com/users/druie/gamepad/
Curves

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

- Higher Degree Approximations
  - Explicit \( y = f(x) \)
  - Implicit \( f(x, y) = 0 \)
  - Parametric \( x = f_x(t), y = f_y(t) \)
Curves

- Explicit Representation
  - \( y = f(x) \quad x \in [a, b] \)
  - 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.,  \( Ax + By + C = 0, x^2 + y^2 - R^2 = 0 \)
  - Harder to render.
  - Simple to check whether a point lies on the curve.
  - Can represent closed and multi-valued curves.
Curves

• Parametric Representation
  - \( x = f_x(t), y = f_y(t) \)

  - Position on the curve is given in terms of a parameter.

  - For e.g., \( x(t) = A(1-t) + Bt, \ y(t) = A(1-t) + Bt, \) with \( t \in [0,1] \)
    \( x(t) = R \cos(t), \ y(t) = R \sin(t), \) with \( t \in [0, 2\pi] \)

  - 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
Cubic Splines

\[ P(t) = B_1 + B_2 t + B_3 t^2 + B_4 t^3 = \sum_{i=1}^{4} B_i t^{i-1} \quad \text{with } t_1 \leq t \leq t_2 \]

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

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

\[ P'(t) = B_2 + 2B_3 t + 3B_4 t^2 = \sum_{i=1}^{4} (i - 1) B_i t^{i-2} \quad \text{with } t_1 \leq t \leq t_2 \]
Cubic Splines

\[ P(t) = B_1 + B_2 t + B_3 t^2 + B_4 t^3 = \sum_{i=1}^{4} B_i t^{i-1} \quad \text{with} \ t_1 \leq t \leq t_2 \]

\[ P'(t) = B_2 + 2B_3 t + 3B_4 t^2 = \sum_{i=1}^{4} (i-1) B_i t^{i-2} \quad \text{with} \ t_1 \leq t \leq t_2 \]

- Given two points and tangent vectors at these two points find a cubic spline that satisfies these end conditions.
- Assuming \( t_1 = 0 \)
  \[ P(0) = P_1, \ P(t_2) = P_2 \]
  \[ P'(0) = P_1', \ P'(t_2) = P_2' \]
Cubic Splines

\[ P(0) = B_1 = P_1 \]

\[ P(t_2) = \sum_{i=1}^{4} B_i t_2^{i-1} = P_2 \]

- On solving we get:

\[ B_1 = P_1 \]

\[ B_2 = P_1' \]

\[ B_3 = \frac{3(P_2 - P_1)}{t_2^2} - \frac{2P_1'}{t_2} - \frac{P_2'}{t_2} \]

\[ B_4 = \frac{2(P_1 - P_2)}{t_2^3} + \frac{P_1'}{t_2^2} + \frac{P_2'}{t_2^2} \]

\[ P'(0) = B_2 = P_1' \]

\[ P'(t_2) = \sum_{i=1}^{4} (i-1) B_i t_2^{i-2} = P_2' \]
Cubic Splines

- Interpolate three points using cubic splines.
- We will do a piecewise polynomial interpolation with some constraints at the join to ensure “smoothness.”
- But what is this notion of smoothness?
Continuity

- **Geometric**
  - $G^0$: Curves are joined
  - $G^1$: First derivatives are proportional. Tangents have same directions but not necessarily the same magnitude
  - $G^2$: First and second derivatives are proportional across the point of joining.

- **Parametric**
  - $C^0$: Curves are joined
  - $C^1$: First derivatives are equal. Tangents have same directions and the same magnitude
  - $C^2$: First and second derivatives are equal across the point of joining.

Parametric continuity of order $n$ implies Geometric continuity, but not vice-versa.
Cubic Splines

- So we enforce $C^2$ continuity at the in-between point.

$$P''(t) = \sum_{i=1}^{4} (i-2)(i-1) B_i t^{i-3}$$

- For the first segment

$$P_1''(t_2) = 6B_4 t_2 + 2B_3, \text{ i.e., at } t=t_2 \text{ with } 0 \leq t \leq t_2$$

- For the second segment

$$P_2''(0) = 2B_3, \text{ i.e., at } t=0 \text{ with } 0 \leq t \leq t_3$$

- So for $C^2$ continuity we have:

$$6B_4 t_2 + 2B_3 = 2B_3$$

Please note the two $B_3$'s in the equation are from different spans.
Cubic Splines

- Substituting and solving we get:

\[ t_3 P_1' + 2(t_3 + t_2) P_2' + t_2 P_3' \]

\[ = \frac{3}{t_2 t_3} \left( t_2^2 (P_3 - P_2) + t_3^2 (P_2 - P_1) \right) \]

\[ \Rightarrow \begin{bmatrix} t_3 & 2(t_3 + t_2) & t_2 \end{bmatrix} \begin{bmatrix} P_1' \\ P_2' \\ P_3' \end{bmatrix} \]

\[ = \frac{3}{t_2 t_3} \left( t_2^2 (P_3 - P_2) + t_3^2 (P_2 - P_1) \right) \]
Cubic Splines

- Similarly for $k^{\text{th}}$ and $k + 1^{\text{th}}$ segments with $1 \leq k \leq n - 2$

\[
\begin{bmatrix}
  t_{k+2} & 2(t_{k+2} + t_{k+1}) & t_{k+1}
\end{bmatrix}
\begin{bmatrix}
  P_k' \\
  P_{k+1}' \\
  P_{k+2}'
\end{bmatrix}
= \frac{3}{t_{k+1} t_{k+2}} \left( t_{k+1}^2 (P_{k+2} - P_{k+1}) + t_{k+2}^2 (P_{k+1} - P_k) \right)
\]

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

\[
\begin{bmatrix}
    t_3 & 2(t_2 + t_3) & t_2 & 0 & \cdots & P_1' \\
    0 & t_4 & 2(t_3 + t_4) & t_3 & \vdots & P_2' \\
    \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\
    \vdots & \ddots & 2(t_{n-1} + t_n) & t_{n-1} & \vdots & P_n' \\
\end{bmatrix}
\]

\[
\begin{align*}
\frac{3}{t_2 t_3} (t_2^2 (P_3 - P_2) + t_3^2 (P_2 - P_1)) \\
\frac{3}{t_3 t_4} (t_3^2 (P_4 - P_3) + t_4^2 (P_3 - P_2)) \\
\vdots \\
\frac{3}{t_{n-1} t_n} (t_{n-1}^2 (P_n - P_{n-1}) + t_n^2 (P_{n-1} - P_{n-2}))
\end{align*}
\]
Cubic Splines

$$\begin{bmatrix}
1 & 0 & \ldots & \\
t_3 & 2(t_2+t_3) & t_2 & 0 & \ldots \\
0 & t_4 & 2(t_3+t_4) & t_3 & \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
t_n & 2(t_{n-1}+t_n) & t_{n-1} & \\
\ldots & \ldots & 0 & 1
\end{bmatrix} =
\begin{bmatrix}
P_1' \\
P_2' \\
\vdots \\
P_{n-1}' \\
P_n'
\end{bmatrix}
$$

\[
\begin{align*}
\frac{3}{t_2 t_3} & \left( t_2^2(P_3 - P_2) + t_3^2(P_2 - P_1) \right) \\
\frac{3}{t_3 t_4} & \left( t_3^2(P_4 - P_3) + t_4^2(P_3 - P_2) \right) \\
& \vdots \\
\frac{3}{t_{n-1} t_n} & \left( t_{n-1}^2(P_n - P_{n-1}) + t_n^2(P_{n-1} - P_{n-2}) \right)
\end{align*}
\]
Cubic Splines

- Solving for \( B_1, B_2, B_3 \) and \( B_4 \)

\[
\begin{align*}
B_{1k} &= P_k \\
B_{2k} &= P_k' \\
B_{3k} &= \frac{3(P_{k+1} - P_k)}{t_{k+1}^2} - \frac{2P_k'}{t_{k+1}} - \frac{2P_{k+1}'}{t_{k+1}} \\
B_{4k} &= \frac{2(P_k - P_{k+1})}{t_{k+1}^3} + \frac{P_k'}{t_{k+1}^2} + \frac{P_{k+1}'}{t_{k+1}^2}
\end{align*}
\]
Cubic Splines

or it can be rewritten as:

$$\begin{bmatrix}
B_{1k} \\
B_{2k} \\
B_{3k} \\
B_{4k}
\end{bmatrix} =
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
-3/t_{k+1} & -2/t_{k+1} & 3/t_{k+1} & -1/t_{k+1} \\
2/t_{k+1} & 1/t_{k+1} & -2/t_{k+1} & 1/t_{k+1}
\end{bmatrix}
\begin{bmatrix}
P_k \\
P_k' \\
P_{k+1} \\
P_{k+1}'
\end{bmatrix}$$
Cubic Splines

Now, for a curve segment of the cubic spline

\[ P_k(t) = \sum_{i=1}^{4} B_{ik} t^{i-1} \] with \(0 \leq t \leq t_{k+1}\) and \(1 \leq k \leq n - 1\)

\[
= \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \begin{bmatrix} B_{1k} \\ B_{2k} \\ B_{3k} \\ B_{4k} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -3/t_{k+1} & -2/t_{k+1} & 3/t_{k+1} & 0 \\ 2/t_{k+1}^3 & 1/t_{k+1} & -2/t_{k+1}^3 & 1/t_{k+1} \end{bmatrix} \begin{bmatrix} P_k \\ P_k' \\ P_{k+1} \\ P_{k+1}' \end{bmatrix}
\]
Cubic Splines

Substitute \( \frac{t}{t_{k+1}} \) as \( \tau \)

\[ P_k(t) = \begin{bmatrix} F_1(\tau) & F_2(\tau) & F_3(\tau) & F_4(\tau) \end{bmatrix} \begin{bmatrix} P_k \\ P_{k+1} \\ P_k' \\ P_{k+1}' \end{bmatrix} \]

where
\[ F_1(\tau) = 2 \tau^3 - 3 \tau^2 + 1 \]
\[ F_2(\tau) = -2 \tau^3 + 3 \tau^2 \]
\[ F_3(\tau) = \tau (\tau^2 - 2 \tau + 1) t_{k+1} \]
\[ F_4(\tau) = \tau (\tau^2 - \tau) t_{k+1} \]

with \( 0 \leq \tau \leq 1 \) and \( 1 \leq k \leq n - 1 \)

These are called the blending or weighting functions
Cubic Splines

\[ P_k(\tau) = F \cdot B \]

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

- \( F_1(0) = 1, F_2(0) = F_3(0) = F_4(0) = 0 \)
  Curve passes through \( P_1 \)

- \( F_1(1) = 0, F_2(1) = 1, F_3(1) = F_4(1) = 0 \)
  Curve passes through \( P_2 \)

- \( F_2(\tau) = 1 - F_1(\tau), F_4(\tau) = -F_3(1 - \tau) \)

- Influence of endpoints constraints vs tangent constraints

CS475/CS675 - Lecture 9
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

\[ \alpha = \pi/4 \]

Figure 5-10 Effect of tangent vector magnitude on cubic spline segment shape, $\alpha = \pi/4$. (a) 1/4; (b) 1/2; (c) 1; (d) 3/cos $\alpha$; (e) 3/2.

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

Figure 5-11  Comparison of cubic spline approximations. (a) Data; (b) connected with straight lines; (c) normalized approximation for $t_k$'s; (d) chord length approximation for $t_k$'s.

Cubic Splines

- Normalized Cubic Splines: Hermite Splines

\[ F_1(\tau) = 2\tau^3 - 3\tau^2 + 1 \]
\[ F_2(\tau) = -2\tau^3 + 3\tau^2 \]
\[ F_3(\tau) = \tau(\tau^2 - 2\tau + 1) \]
\[ F_4(\tau) = \tau(\tau^2 - \tau) \]

or

\[ \begin{bmatrix} F_1(\tau) & F_2(\tau) & F_3(\tau) & F_4(\tau) \end{bmatrix} \]

- The tridiagonal system becomes:

\[
\begin{bmatrix}
1 & 0 & \cdots & \\
1 & 4 & 1 & \cdots \\
\vdots & \ddots & \ddots & \vdots \\
\cdots & 1 & 4 & 1 \\
\cdots & 0 & 1 & \cdots \\
\end{bmatrix}
\begin{bmatrix}
P_1' \\
P_2' \\
\vdots \\
P_{n-1}' \\
P_n' \\
\end{bmatrix}
= 
\begin{bmatrix}
P_1' \\
3((P_3-P_2)+(P_2-P_1)) \\
3((P_4-P_3)+(P_3-P_2)) \\
\vdots \\
3((P_n-P_{n-1})+(P_{n-1}-P_{n-2})) \\
P_n' \\
\end{bmatrix}
\]
Cubic Splines

- Various end conditions for Cubic Splines
  - Clamped: $P_1'(0) = P_1'(0)$ and $P_n'(t_n) = P_n'(0)$ 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 $P_1''(0) = -P_n''(t_n)$