Natural Splines

- Draw a "smooth" line through several points

A real draftsman's spline.
Image from Carl de Boor's webpage.
Natural Cubic Splines

• Given \( n + 1 \) points
  • Generate a curve with \( n \) segments
  • Curves passes through points
  • Curve is \( C^2 \) continuous

• Use cubics because lower order is better...

\[
x(u) = \begin{cases} 
  s_0(u) & \text{if } 0 \leq u < 1 \\
  s_1(u - 1) & \text{if } 1 \leq u < 2 \\
  s_2(u - 2) & \text{if } 2 \leq u < 3 \\
  \vdots \\
  s_n(u - (n - 1)) & \text{if } n - 1 \leq u \leq n 
\end{cases}
\]
Natural Cubic Splines

- Interpolate data points
- No convex hull property
- Non-local support
  - Consider matrix structure...
  - $C^2$ using cubic polynomials

\[
\begin{align*}
  s_i(0) &= p_{i-1} & i = 1 \ldots n & \text{N constraints} \\
  s_i(1) &= p_i & i = 1 \ldots n & \text{N constraints} \\
  s_i'(1) &= s_{i-1}'(0) & i = 1 \ldots n - 1 & \text{N-1 constraints} \\
  s_i'(1) &= s_{i+1}'(0) & i = 1 \ldots n - 1 & \text{N-1 constraints} \\
  s_i''(0) &= s_i''(1) = 0 & \text{2 constraints} \\
  s_0''(1) &= 0 & \text{2 constraints} \\
\end{align*}
\]

Total 4N constraints
B-Splines

• Goal: \( c^2 \) cubic curves with local support
  • Give up interpolation
  • Get convex hull property
• Build basis by designing "hump" functions

\[
b(u) = \begin{cases} b_{3}(u) & \text{if } u_{2} \leq u < u_{1} \\
b_{2}(u) & \text{if } u_{1} \leq u < u_{0} \\
b_{1}(u) & \text{if } u_{0} \leq u < u_{1} \\
b_{0}(u) & \text{if } u_{1} \leq u \leq u_{2} \\
b_{+1}(u) & \text{if } u_{0} \leq u < u_{1} \\
b_{+2}(u) & \text{if } u_{1} \leq u \leq u_{2} \\
b_{00}(u) & \text{if } u_{2} \leq u < u_{1} \\
b_{01}(u) & \text{if } u_{1} \leq u < u_{0} \\
b_{02}(u) & \text{if } u_{0} \leq u < u_{1} \\
b_{+1}(u) & \text{if } u_{1} \leq u \leq u_{2} \\
b_{+2}(u) & \text{if } u_{0} \leq u < u_{1} \\
b_{+3}(u) & \text{if } u_{1} \leq u \leq u_{2} \\
0 & \text{else}
\end{cases}
\]

Total 15 constraints ... need one more
B-Splines

\[ b(u) = \begin{cases} 
  b_{i,0}(u) & \text{if } u_{i-1} \leq u < u_i \\
  b_{i,1}(u) & \text{if } u_{i-1} \leq u < u_i \\
  b_{i,2}(u) & \text{if } u_{i-1} \leq u < u_i
\end{cases} \]

\[ b_{i,j}(u_{i+2}) = b_{i,j}(u_{i-2}) = 0 \quad \text{← 3 constraints} \]

\[ b_{i,j}(u_{i+2}) = b_{i,j}(u_{i-2}) = 0 \quad \text{← 3 constraints} \]

\[ \begin{align*}
  b_{i,j}(u_i) &= b_{i,j}(u_i) \\
  b_{i,j}(u_i) &= b_{i,j}(u_i) \\
  b_{i,j}(u_i) &= b_{i,j}(u_i)
\end{align*} \quad \text{Repeat for } y' \text{ and } y'' \]

\[ 3 \times 3 = 9 \text{ constraints} \]

\[ b_{i,j}(u_{i+2}) + b_{i,j}(u_{i+1}) + b_{i,j}(u_{i}) + b_{i,j}(u_{i+2}) = 1 \quad \text{← 1 constraint (convex hull)} \]

Total 16 constraints
Example with end knots repeated
<table>
<thead>
<tr>
<th>B-Splines</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Build a curve w/ overlapping bumps</td>
</tr>
<tr>
<td>• Continuity</td>
</tr>
<tr>
<td>• Inside bumps $C^2$</td>
</tr>
<tr>
<td>• Bumps “fade out” w/ $C^2$ continuity</td>
</tr>
<tr>
<td>• Boundaries</td>
</tr>
<tr>
<td>• Circular</td>
</tr>
<tr>
<td>• Repeat end points</td>
</tr>
<tr>
<td>• Extra end points</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B-Splines</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Notation</td>
</tr>
<tr>
<td>• The basis functions are the $B_i(u)$</td>
</tr>
<tr>
<td>• “Hump” functions are the concatenated function</td>
</tr>
<tr>
<td>• Sometimes the humps are called basis, can be confusing</td>
</tr>
<tr>
<td>• The $u_i$s are the knot locations</td>
</tr>
<tr>
<td>• The weights on the hump/basis functions are control points</td>
</tr>
</tbody>
</table>
B-Splines

• Similar construction method can give higher continuity with higher degree polynomials
• Repeating knots drops continuity
  • Limit as knots approach each other
• Still cubics, so conversion to other cubic basis is just a matrix multiplication

B-Splines

• Geometric construction
  • Due to Cox and de Boor
  • My own notation, beware if you compare w/ text!
• Let hump centered on $u_i$ be $N_{i,k}(u)$
  Cubic is order 4

$N_{i,k}(u)$ is order $k$ hump, centered at $u_i$

Note: $i$ is integer if $k$ is even
else $(i + 1/2)$ is integer
B-Splines
NURBS

- Non-uniform Rational B-Splines
  - Basically B-Splines using homogeneous coordinates
  - Transform under perspective projection
  - A bit of extra control

\[ x(u) = \frac{\sum_i p_{iw} N_i(u)}{\sum_i p_{iw} N_i(u)} \]

- Non-linear in the control points
- The \( p_{iw} \) are sometimes called “weights”