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 pass through points
  - Curve is $C^2$ continuous

- Use cubics because lower order is better...

$x(u) = \begin{cases} 
  s_1(u) & \text{if } 0 \leq u < 1 \\
  s_2(u - 1) & \text{if } 1 \leq u < 2 \\
  s_3(u - 2) & \text{if } 2 \leq u < 3 \\
  \vdots & \\
  s_n(u - (n - 1)) & \text{if } n - 1 \leq u \leq n 
\end{cases}$
Natural Cubic Splines

\[ s_i(0) = p_{i-1} \quad i = 1 \ldots n \]
\[ s_i(1) = p_i \quad i = 1 \ldots n \]
\[ s_i''(1) = s_{i+1}''(0) \quad i = 1 \ldots n - 1 \]
\[ s_i''(0) = s_{n-1}''(1) = 0 \]

\[ \leftarrow n \text{ constraints} \]
\[ \leftarrow n \text{ constraints} \]
\[ \leftarrow n-1 \text{ constraints} \]
\[ \leftarrow n-1 \text{ constraints} \]
\[ \leftarrow 2 \text{ constraints} \]

**Total 4n constraints**

Interpolate data points
No convex hull property
Non-local support
Consider matrix structure...
\[ C^2 \text{ using cubic polynomials} \]
B-Splines

- **Goal:** $C^2$ cubic curves with local support
  - Give up interpolation
  - Get convex hull property
  - Build basis by designing “hump” functions

\[ b(u) = \begin{cases} 
  b_{-2}(u) & \text{if } u_{-2} \leq u < u_{-1} \\
  b_{-1}(u) & \text{if } u_{-1} \leq u < u_0 \\
  b_{+1}(u) & \text{if } u_0 \leq u < u_{+1} \\
  b_{+2}(u) & \text{if } u_{+1} \leq u \leq u_{+2} 
\end{cases} \]

$b''_{-2}(u_{-2}) = b''_{-2}(u_{-1}) = b''_{-2}(u_{-2}) = 0$ \leftarrow 3 constraints

$b''_{+2}(u_{+2}) = b''_{+2}(u_{+2}) = b''_{+2}(u_{+2}) = 0$ \leftarrow 3 constraints

- \[ b_{-2}(u_{-1}) = b_{-1}(u_{-1}) \]
- \[ b_{-1}(u_0) = b_{+1}(u_0) \]
- \[ b_{+1}(u_{+1}) = b_{+2}(u_{+1}) \]

Repeat for $b'$ and $b''$

3x3=9 constraints

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

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

\[ b''_{-2}(u_{-2}) = b'_{-2}(u_{-2}) = b_{-2}(u_{-2}) = 0 \quad \leftarrow 3 \text{ constraints} \]
\[ b''_{+2}(u_{+2}) = b'_{+2}(u_{+2}) = b_{+2}(u_{+2}) = 0 \quad \leftarrow 3 \text{ constraints} \]

\[ b_{-2}(u_{-1}) = b_{-1}(u_{-1}) \]
\[ b_{-1}(u_0) = b_{+1}(u_0) \]
\[ b_{+1}(u_{+1}) = b_{+2}(u_{+1}) \]

\[ b_{-2}(u_{-2}) + b_{-1}(u_{-1}) + b_{+1}(u_0) + b_{+2}(u_{+1}) = 1 \quad \leftarrow 1 \text{ constraint (convex hull)} \]

Total 16 constraints

B-Splines

- Build a curve w/ overlapping bumps
- Continuity
  - Inside bumps \( C^2 \)
  - Bumps “fade out” with \( C^2 \) continuity
- Boundaries
  - Circular
  - Repeat end points
  - Extra end points
B-Splines

- **Notation**
  - The basis functions are the $b_i(u)$
  - “Hump” functions are the concatenated function
    - Sometimes the humps are called basis... can be confusing
  - The $u_i$ are the knot locations
  - The weights on the hump/basis functions are control points

- 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
Note: \( i \) is integer if \( k \) is even else \((i+1/2) \) is integer

Due to Cox and de Boor
My own notation, beware if you compare w/ text

B-Splines

NURBS

- **Nonuniform Rational B-Splines**
  - Basically B-Splines using homogeneous coordinates
  - Transform under perspective projection
  - A bit of extra control
Non-linear in the control points

The \( p_{iw} \) are sometimes called “weights”