Tomorrow, Thursday, 10:45, study session on Zoom. All are welcome.

Review

- polynomial interpolant: $C^\infty$, global, error may not converge to zero.
- piecewise linear interpolant: only $C^0$, local, Error goes to zero like $O(h^2)$.
- piecewise cubic Hermite interpolant: $C^1$, local, Error goes to zero like $O(h^4)$, requires derivative values.

Cubic Splines

- This is another classic subject. Here are two classic references:
- A recent Matlab oriented text is:
• Again, assume we are given data \((x_i, y_i), i = 0, \ldots, n\) where the knots are ordered:

\[
a = x_0 < x_1 < \ldots < x_n = b
\]

• The word “spline” derives from a mechanical gadget that used to be used to draw smooth curves.

• Pass an elastic wire through some points and let it adjust itself so that it minimizes its strain energy.
• The curvature of the curve $y = s(x)$ is

$$\kappa(x) = \left| \frac{s''}{(1 + s'^2)^{3/2}} \right|.$$ 

• If $s'$ is small (or close to constant) the curvature is approximately proportional to $s''$.

• So the mechanical idea can be approximated by the mathematical problem of finding the interpolating function $s$ that interpolates, i.e.,

$$s(x_i) = y_i, \quad i = 0, 1, 2, \ldots, n$$

and that minimizes the integral

$$\int_a^b (s''(x))^2 \, dx = \min$$

• It is a great exercise to show that the solution of this variational problem is a cubic spline, i.e., a function that is cubic on each subinterval

$$I_i = [x_{i-1}, x_i]$$

and that is twice differentiable on $[a, b]$. 
Count parameters and conditions

\[ s(x) \text{ piecewise cubic} \]

\[ x_0 \quad x_1 \quad \ldots \quad x_n \]

n subintervals

\[ 4n \text{ # parameters} \]

\[ \text{# conds.} \]

\[ s(x_i) = y_i; \quad i = 0, \ldots, n \]

\[ 2(n-1) + 2 = 2n \]

\[ s \in \mathbb{C}^{(1)} \]

\[ \begin{array}{c}
\frac{n-1}{n-1}
\\hline
\end{array} \]

\[ 4n - 2 \text{ conds.} \]

\[ 4n \text{ params.} \]
Boundary Conditions

• Three kinds of boundary conditions are in frequent use:

1. **natural end conditions.** Let

\[ s''(a) = s''(b) = 0 \]

• This is called **natural** since the above discussed wire would be linear to the left of \( a \) and to the right of \( b \).

2. **Forced End Condition.** Let

\[ s'(a) = A, \quad s'(b) = B. \]

This means forcing the wire to point in certain directions at the endpoints. The obvious question is how to pick \( A \) and \( B \).

3. **Not-a-Knot condition.** Force the spline to be three times differentiable at \( x_1 \) and \( x_{n-1} \):

\[
\begin{align*}
\lim_{x \to x_1^-} s'''(x) &= \lim_{x \to x_1^+} s'''(x) \\
\lim_{x \to x_{n-1}^-} s'''(x) &= \lim_{x \to x_{n-1}^+} s'''(x)
\end{align*}
\]

• A piecewise cubic function that is three times differentiable is actually cubic, so \( x_1 \) and \( x_{n-1} \) are not knots. The resulting spline is cubic, rather than piecewise cubic, on \([x_0, x_2]\) and \([x_{n-2}, x_n]\).
• A **Cardinal Spline** is a spline satisfying the cardinal condition

\[ s(x_i) = \delta_{ij} \]

The support of a cardinal spline is the entire interval \([a, b]\) (except at the knots). Geometrically, if you move any one data point the wire wiggles everywhere.

• This means that **Spline interpolation is global.**
B-splines

- **B-splines** are splines with support that is small as possible, which for cubic splines means 4 intervals.

- B-splines form a large subject.
Polynomial Precision

- We say that an interpolation scheme has **polynomial precision** $k$ if the interpolant of a polynomial of degree up to $k$ is that polynomial.

- We also say that the interpolation scheme reproduces polynomials of degree up to $k$.

- Polynomial interpolation of $n+1$ data reproduces polynomials of degree up to

- Piecewise linear interpolation reproduces polynomials of degree up to

- Piecewise Cubic Hermit interpolation reproduces polynomials of degree up to

- What about cubic spline interpolation:

  **Exercise**
Computation of Cubic Splines

- We only have to solve a tridiagonal linear system!


- Exercise: Verify everything and work out the details!

- The second derivative of a cubic spline $s$ is piecewise linear and continuous. Letting

$$ M_i = s''(x_i) \quad \text{and} \quad I_i = [x_{i-1}, x_i] $$

we set

$$ h_i = x_i - x_{i-1} $$

as before and get for $x \in I_i$:

$$ s''(x) = M_{i-1} \frac{x_i - x}{h_i} + M_i \frac{x - x_{i-1}}{h_i}. $$

- Integrating $s''$ twice gives

$$ s(x) = M_{i-1} \frac{(x_i - x)^3}{6h_i} + M_i \frac{(x - x_{i-1})^3}{6h_i} + c_i(x_i - x) + d_i(x - x_{i-1}) $$

where the $c_i$ and $d_i$ are as yet undetermined integration constants.
• We must have

\[ s(x_{i-1}) = y_{i-1} \quad \text{and} \quad s(x_i) = y_i. \]

• This determines the constants \( c_i \) and \( d_i \) and gives (still for \( x \in I_i \))

\[
\begin{align*}
  s(x) &= M_{i-1} \frac{(x_i - x)^3}{6h_i} + M_i \frac{(x - x_{i-1})^3}{6h_i} \\
  &\quad + \left( y_{i-1} - \frac{M_{i-1} h_i^2}{6} \right) \frac{x_i - x}{h_i} \\
  &\quad + \left( y_i - \frac{M_i h_i^2}{6} \right) \frac{x - x_{i-1}}{h_i}
\end{align*}
\]  

(1)

• Exercise: compute the \( c_i \) and \( d_i \) in terms of the \( M_j, h_j \) and \( y_j \).

• We still need to define the \( M_i \). They can be obtained from the \( C^1 \) conditions

\[
\lim_{x \to x_i^-} s'(x) = \lim_{x \to x_i^+} s'(x).
\]

• Differentiating in (1) gives:

\[
\begin{align*}
  S'(x) &= -M_{i-1} \frac{(x_i - x)^2}{2h_i} + M_i \frac{(x - x_{i-1})^2}{2h_i} \\
  &\quad + \frac{y_i - y_{i-1}}{h_i} + \frac{M_{i-1} - M_i}{6} \frac{1}{h_i}
\end{align*}
\]  

(2)
• Evaluating (2) and the expression on the next interval at \( x_i \) and equating the two expression gives the equation

\[
\frac{h_i}{2} M_i + \frac{M_{i-1} - M_i}{6} h_i + \frac{y_i - y_{i-1}}{h_i} = \\
- M_i \frac{h_{i+1}}{2} + \frac{y_{i+1} - y_i}{h_{i+1}} + \frac{M_i - M_{i+1}}{6} h_{i+1}
\]

which can be rewritten as

\[
\alpha_i M_{i-1} + \beta_i M_i + \gamma_i M_{i+1} = \delta_i
\]

where

\[
\alpha_i = \frac{h_i}{6} \\
\beta_i = \frac{1}{3} (h_i + h_{i+1}) \\
\gamma_i = \frac{h_{i+1}}{6} \\
\delta_i = \frac{y_{i-1} - y_i}{h_i} + \frac{y_{i+1} - y_i}{h_{i+1}}
\]

\[
i = 1, 2, \ldots, n - 1
\]

These are \( n - 1 \) equations in \( M_0, \ldots, M_n \).

• The two missing conditions are the boundary conditions. For example, natural splines have the boundary conditions

\[
M_0 = M_n = 0.
\]
Adding these as the first and last equations to (3) gives a square tridiagonal \((n+1) \times (n+1)\) tridiagonal linear system.

- Exercise: work out the missing equations for forced end and knot-a-knot conditions.