Interpolation

• Construct a function crosses known points
• Predict the value of unknown points
Interpolation in modeling
Interpolation

• Polynomial Interpolation
  – Same polynomial for all points
  – Vandermonde Matrix, ill-conditioned

• Lagrange Form
  – Hard to evaluate

• Piecewise Interpolation
  – Different polynomials for each interval
Lagrange form

Given \( k+1 \) points

\((x_0, y_0), \ldots, (x_j, y_j), \ldots, (x_k, y_k)\)

Define:

\[
L(x) := \sum_{j=0}^{k} y_j \ell_j(x)
\]

where

\[
\ell_j(x) := \prod_{0 \leq m \leq k \atop m \neq j} \frac{x - x_m}{x_j - x_m} = \frac{(x - x_0)}{(x_j - x_0)} \cdots \frac{(x - x_{j-1})}{(x_j - x_{j-1})} \frac{(x - x_{j+1})}{(x_j - x_{j+1})} \cdots \frac{(x - x_k)}{(x_j - x_k)}.
\]
Lagrange form

\[ \ell_j(x_i) = \delta_{ji} = \begin{cases} 1, & \text{if } j = i \\ 0, & \text{if } j \neq i \end{cases} \]

\[ L(x_i) = \sum_{j=0}^{k} y_j \ell_j(x_i) = \sum_{j=0}^{k} y_j \delta_{ji} = y_i. \]
Lagrange form

• Example: interpolate \( f(x) = x^2 \), for \( x = 1, 2, 3 \)

\[
x_0 = 1 \quad f(x_0) = 1 \\
x_1 = 2 \quad f(x_1) = 4 \\
x_2 = 3 \quad f(x_2) = 9.
\]

\[
L(x) = 1 \cdot \frac{x - 2}{1 - 2} \cdot \frac{x - 3}{1 - 3} + 4 \cdot \frac{x - 1}{2 - 1} \cdot \frac{x - 3}{2 - 3} + 9 \cdot \frac{x - 1}{3 - 1} \cdot \frac{x - 2}{3 - 2} = x^2.
\]
How to represent models

• Specify every point along a model?
  – Hard to get precise results
  – Too much data, too hard to work with generally

• Specify a model by a small number of “control points”
  – Known as a spline curve or just spline
Spline Interpolation

• For some cases, polynomials can lead to erroneous results because of round off error and overshoot.

• Alternative approach is to apply lower-order polynomials to subsets of data points. Such connecting polynomials are called spline functions.
Spline Interpolation Definition

- Given \( n+1 \) distinct knots \( x_i \) such that:
  \[
  x_0 < x_1 < \ldots < x_{n-1} < x_n,
  \]
  with \( n+1 \) knot values \( y_i \) find a spline function

\[
S(x) := \begin{cases} 
  S_0(x) & x \in [x_0, x_1] \\
  S_1(x) & x \in [x_1, x_2] \\
  \vdots \\
  S_{n-1}(x) & x \in [x_{n-1}, x_n]
\end{cases}
\]

with each \( S_i(x) \) a polynomial of degree at most \( n \).
Tangent

- The derivative of a curve represents the tangent vector to the curve at some point.
(a) Linear spline
   – Derivatives are not continuous
   – Not smooth

(b) Quadratic spline
   – Continuous 1\(^{st}\) derivatives

(c) Cubic spline
   – Continuous 1\(^{st}\) & 2\(^{nd}\) derivatives
   – Smoother
• Why cubic?
  – Good enough for some cases
  – The degree is not too high to be easily solved
Natural Cubic Spline Interpolation

- \( S_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i \) (Given \( n \) points)
  - 4 Coefficients with \( n-1 \) subintervals = 4n-4 equations
  - There are 4n-6 conditions
    - Interpolation conditions
    - Continuity conditions
    - Natural Conditions
  - \( S''(x_0) = 0 \)
  - \( S''(x_n) = 0 \)
  - \( O(n^3) \)
Natural Cubic Spline Interpolation

- A clever method
  - Construct $S(x)$
    - Lagrange Form thought
  - Solve tridiagonal matrix
    - Using decompt & solvet (2-1)
  - Evaluate of $S(z)$
    - Locate $z$ in some interval (using binary search)
    - Using Horner’s rule to evaluate
Thanks