CSC349A Numerical Analysis

Lecture 16

Rich Little

University of Victoria

2023

Table of Contents I

Piecewise interpolation

University of Victoria Computer Science

An alternative to polynomial interpolation use "piecewise" polynomials.
Given $x_0, x_1, \ldots, x_n$ and $f(x_0), f(x_1), \ldots, f(x_n)$ construct a different interpolating polynomial on each subinterval:

$$[x_0, x_1], [x_1, x_2], \ldots, [x_{n-1}, x_n]$$

For example piecewise linear interpolation: construct a linear polynomial on each subinterval $[x_i, x_{i+1}]$.

Linear splines

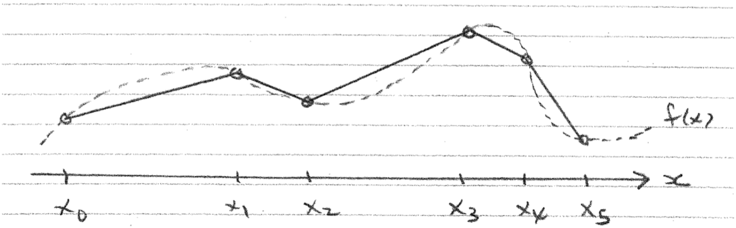

Figure: Example of linear spline

Disadvantage of piecewise linear polynomials: not differentiable (at points x_i, the knots).

Quadratic splines

Differentiability can be obtained by using quadratic (instead of linear) polynomials on each $[x_i, x_{i+1}]$.

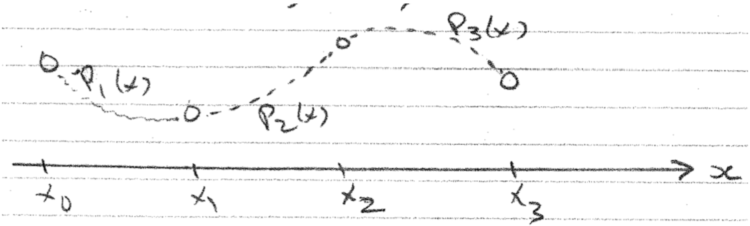

Figure: Example of quadratic spline

Quadratic splines

- Each $P_i(x)$ is a quadratic (and is not uniquely determined)
- The piecewise polynomial can be made differentiable on $[x_0, x_n]$
- If differentiable, this is an example of a <u>spline function</u>

Spline Defintion

Definition: $S(x)$ is a spline function on $[x_0, x_n]$ if for some $q \geq 1$

1. $S(x)$ is a polynomial of degree q on each subinterval $[x_i, x_{i+1}]$
2. $S(x)$ and its first $q - 1$ derivatives are continuous on $[x_0, x_n]$

Spline types:

- Linear spline, $q = 1$
- Quadratic spline, $q = 2$
- Cubic spline, $q = 3$

Physical splines

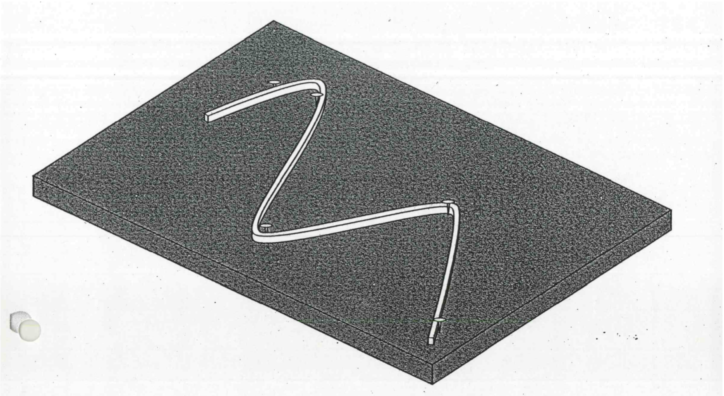

Figure: Drafting technique of using a spline to draw smooth curves through a series of points

History

Splines were first defined by Schoenberg in 1946. Note that the definition of a spline function does **not** require that it interpolates some given function $f(x)$. But splines are often used as interpolating functions (a spline interpolant):

- They do not have the oscillatory nature of high degree interpolating polynomials
- They require no derivatives of $f(x)$, except possibly at the end points x_0 and x_n.

The most common spline interpolant is **cubic**.

Applications

- Graphics
  - Smooth curves (continuity)
- Animation
  - Modeling = specifying shape
  - Animation = specifying shape over time
  - Real objects don't move in straight lines - Video
- Motion Control
  - embedded systems
  - automated motion
  - robotics

Table of Contents I

Cubic Spline Interpolants

Definition: Given $x_0, x_1, ..., x_n$ with $x_i < x_{i+1}$ for each i, and $f(x_0), f(x_1), ..., f(x_n)$, then $S(x)$ is a **cubic spline interpolant** for $f(x)$ if,

(a) $S(x)$ is a cubic polynomial, denoted by $S_j(x)$, on each subinterval $[x_j, x_{j+1}]$, for $j = 0, ..., n-1$
(b) $S_j(x_j) = f(x_j)$, for $j = 0, ..., n-1$ and $S_{n-1}(x_n) = f(x_n)$
(c) $S_{j+1}(x_{j+1}) = S_j(x_{j+1})$, for $j = 0, ..., n-2$
(d) $S'_{j+1}(x_{j+1}) = S'_j(x_{j+1})$, for $j = 0, ..., n-2$
(e) $S''_{j+1}(x_{j+1}) = S''_j(x_{j+1})$, for $j = 0, ..., n-2$
(f) either one of the following hold:
  (i) $S''(x_0) = S''(x_n) = 0$ (natural bounds), or
  (ii) $S'(x_0) = f'(x_0)$ and $S'(x_n) = f'(x_n)$ (clamped bounds)

Cubic Spline Interpolants II

Notes:

- for any $f(x)$, there exist an infinite number of cubic splines satisfying conditions (a) - (e). Why?
- There are n cubic polynomials $S_j(x)$ to specify, each one is defined by 4 coefficients, giving a total of $4n$ unknowns to be specified.
- However, condition (b) gives $n+1$ conditions to be satisfied, and (c), (d) and (e) each give $n-1$ conditions to be satisfied.
- Thus, there are $(n+1)+3(n-1)=4n-2$ conditions (equations) to be satisfied in $4n$ unknowns.
- But if either (i) or (ii) is also required to be satisfied, then there are $4n$ conditions in $4n$ unknowns and there exists a <u>unique</u> cubic spline interpolant satisfying (a) - (f).

Example 1 - Cubic Spline

Determine $a_0, b_0, d_0, a_1, b_1, c_1,$ and d_1 so that

$$S(x) = \begin{cases} a_0 + b_0 x - 3x^2 + d_0 x^3, & -1 \le x \le 0 \\ a_1 + b_1 x + c_1 x^2 + d_1 x^3, & 0 \le x \le 1 \end{cases}$$

is the natural cubic spline function such that
$S(-1) = 1, S(0) = 2, S(1) = -1.$

Example 1 continued

Example 1 continued

Cubic Splines in MATLAB

There is an algorithm for spline computation given in the text but it has a different derivation than what we have done and different from MATLAB. In MATLAB they use a different form for the splines. For example, when $n = 3$, MATLAB uses the following form for the cubic polynomials:

$$
\begin{aligned}
S_0(x) &= a_0 + b_0(x - x_0) + c_0(x - x_0)^2 + d_0(x - x_0)^3 \\
S_1(x) &= a_1 + b_1(x - x_1) + c_1(x - x_1)^2 + d_1(x - x_1)^3 \\
S_2(x) &= a_2 + b_2(x - x_2) + c_2(x - x_2)^2 + d_2(x - x_2)^3
\end{aligned}
$$

Note that, with this form, $a_0 = f(x_0)$, $a_1 = f(x_1)$, and $a_2 = f(x_2)$. This simplifies the system somewhat.

Quadratic Spline

Construction of quadratic splines is similar to that of cubic splines but there are only $3n$ unknown coefficients and you do not need to set the second derivatives of the interior knots to be equal. That is, we do not create the (e) equations from above. As such, the (b) to (d) equations total $3n-1$, meaning that we also only need one extra (f) equation. Often we use $Q''(x_0) = 0$ or, as is the case with the next example, we assign one of the coefficients before hand.

Example 2 - Quadratic Spline

Determine a, b, c, d, and e so that

$$Q(x) = \begin{cases} ax^2 + x + b, & -1 \leq x \leq 0 \\ cx^2 + dx + e, & 0 \leq x \leq 1 \end{cases}$$

is a quadratic spline function that interpolates $f(x)$ where $f(-1) = 1, f(0) = 1, f(1) = 1$.

Example 2 continued