Natural cubic splines:

Splines:

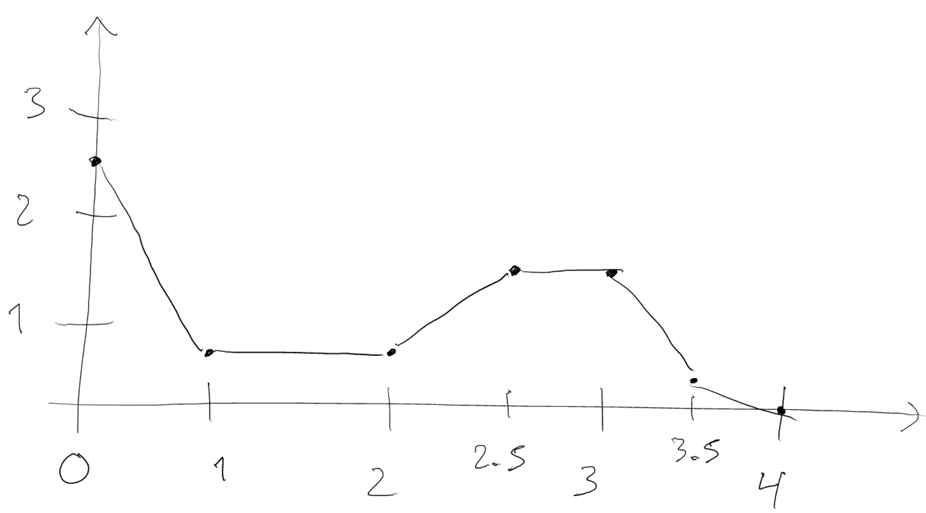
- Used to render scalable fonts
- Used for animation and computer graphics, CAD

They're useful whenever you need mathematical representations of free form curved surfaces.

Suppose you want to interpolate the data

x	0	1	2	2.5	3	3.5	4
y	2.5	0.5	0.5	1.5	1.5	1.125	0

Simplest method of interpolation is to connect node points with line segments.
(plot(x,y) in MATLAB does this.)
This is called piecewise linear interpolation.

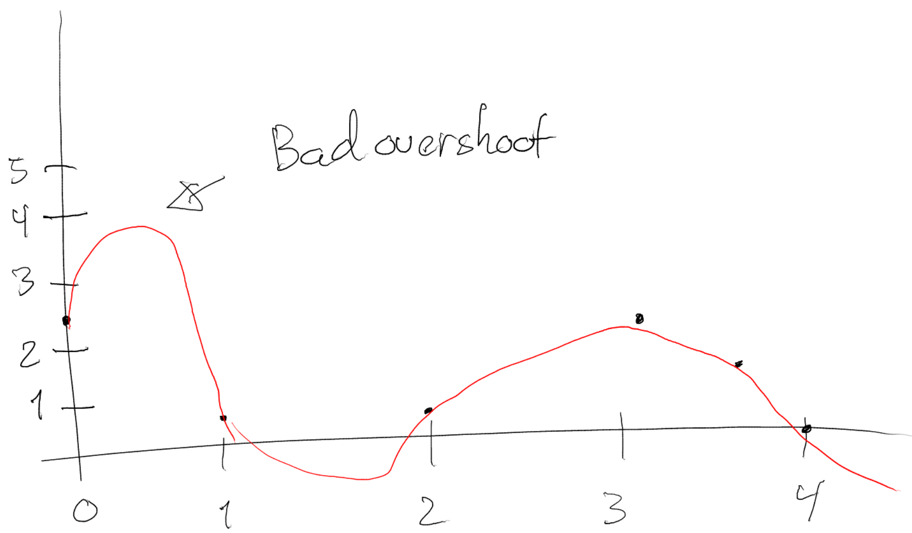


Good: Fits data

Bad: Not smooth.

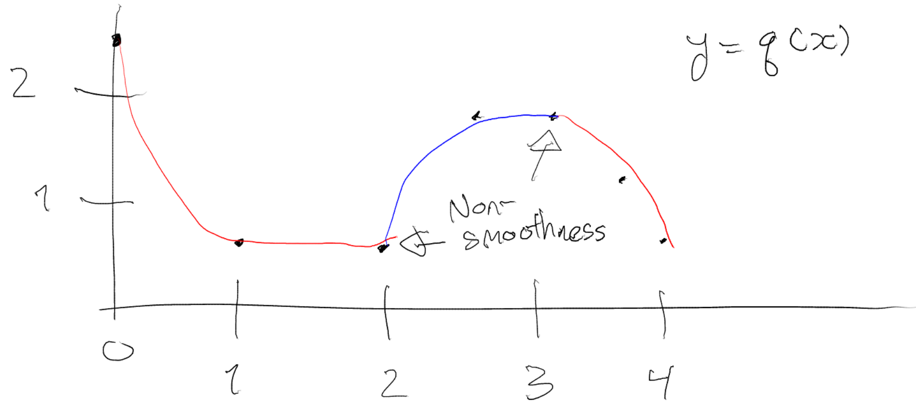
For visualization and other applications, we often want to construct a smooth curve that interpolates the data and is similar/close to this piecewise linear interpolation.

Next option might be to use polynomial interpolation to fit a degree 6 polynomial to our 7 data points.



Not close to the piecewise linear function.

For the third option, we may connect the data points using a collection of quadradic interpolating polynomial on subintervals of 3-data points at a time.



This function is smoother than the piecewise linear one.
It follows the piecewise linear one closer than polynomial interpolation does.
But at points $x=2$ and $x=3$, you still get lack of smoothness.

<u>Goal</u>: Find an interpolating function that is smooth and doesn't change too much between the node points (similar to piecewise linear).

<u>Three criteria</u>: Given N data points $\{(x_i, y_i)\}_{i=1}^{N}$ seek $y = s(x)$ so that

1. $s(x_i) = y_i, \quad i = 1, 2, \dots, N$
2. s', s'' to be continuos
3. Want function to be close to piecewise linear interpolation.

To ensure 3, we ask that s' not change too much between node points.

The most common choice of polynomial degree to use is 3, i.e. cubic splines.
We'll join cubic polynomials together so that the resulting spline function verifies 1 - 3.

(There's no real advantage and some disadvantages with spline degree > 3).

The unique solution $s(x)$ should satisfy

① s is a polynomial of degree ≤ 3 on each subinterval $[x_{j-1}, x_j]$, $j = 2, 3, \ldots, N$

② s, s', s'' are continuos on entire interval $[a, b]$

③ $s''(x_1) = 0 = s''(x_n)$

Condition ③ is included to ensure we get the same number of equations as unknown coeffecients in all the cubics so that we have a chance to get a unique solution.

This s is called the natural cubic spline that interpolates our data.

<u>Algorithms for natural cubic splines</u>: Data (t_i, y_i)

Set $z_i = s''(t_i)$, $\forall i$ (unknown for now).

On the interval $[t_i, t_{i+1}]$, s is cubic. So s'' is linear. We know

$$s''(t_i) = z_i, \quad s''(t_{i+1}) = z_{i+1}.$$

So for $x \in [t_i, t_{i+1}]$, we have

$$s_i''(x) = z_{i+1} \frac{x - t_i}{h_i} + z_i \frac{t_{i+1} - x}{h_i}$$

where $h_i = t_{i+1} - t_i$. Integrate twice to get

$$s_i(x) = \frac{z_{i+1}}{6h_i}(x - t_i)^3 + \frac{z_i}{6h_i}(t_{i+1} - x)^3 + cx + d$$

It's better to rewrite coeffecients of integration.

$$S_i(x) = \frac{z_{i+1}}{6h_i}(x-t_i)^3 + \frac{z_i}{6h_i}(t_{i+1}-x)^3 + C_i(x-t_i) + D_i(t_{i+1}-x)$$

To solve for C_i and D_i plug into $S_i(t_i)=y_i$, $S_i(t_{i+1})=y_{i+1}$.

Then,

$$y_i = \frac{z_i h_i^2}{6} + D_i h_i, \quad \text{from } S_i(t_i)=y_i.$$

So,

$$D_i = \frac{y_i}{h_i} - \frac{z_i h_i}{6}.$$

We know h_i and y_i but not z_i.

Similarly,

$$C_i = \frac{y_{i+1}}{h_i} - \frac{z_{i+1} h_i}{6}.$$

We know h_i and y_i but not z_{i+1}.

<u>Upshot:</u>

$$S_i(x) = \frac{z_{i+1}}{6h_i}(x-t_i)^3 + \frac{z_i}{6h_i}(t_{i+1}-x)^3 + \left(\frac{y_{i+1}}{h_i} - \frac{h_i}{6}z_{i+1}\right)(x-t_i) + \left(\frac{y_i}{h_i} - \frac{h_i}{6}z_i\right)(t_{i+1}-x).$$

Just need to solve for z_i, z_{i+1}. We have not imposed yet the condition

$$S_{i-1}'(t_i) = S_i'(t_i), \quad \forall i.$$

You can check

$$S_i'(t_i) = \frac{y_{i+1}}{h_i} - \frac{h_i z_{i+1}}{6} - \frac{y_i}{h_i} - \frac{z_i h_i}{3} \qquad (1)$$

$$S_{i-1}'(t_i) = \frac{h_{i-1}}{6} z_{i-1} + \frac{h_{i-1} z_i}{3} + \frac{y_i}{h_{i-1}} - \frac{y_{i-1}}{h_{i-1}} \qquad (2)$$

Setting (1) = (2) for all i gives us a tridiagonal system of linear equation for the z's.

$z_0 = 0$ from criteria (3).

$h_{i-1} z_{i-1} + u_i z_i + h_i z_{i+1} = v_i \qquad i = 1, \dots, N-1$

$z_n = 0$ from criteria (3)

where

N+1 equations

$b_i = \frac{1}{h_i}(y_{i+1} - y_i)$ ← finite diffence on the data

$u_i = 2(h_{i-1} + h_i)$

$v_i = 6(b_i - b_{i-1})$ ← Some sort of second difference

Example: Calculate the natural cubic spline to interpolate the data

x	-1	0	1
y	1	2	-1

$(N=2)$

We have 3 node points t_0, t_1, t_2.

Our tridiagonal system is 3x3:

$$\begin{cases} z_0 = 0 \\ h_0 z_0 + u_1 z_1 + h_1 z_2 = v_1 \\ z_2 = 0 \end{cases}$$

$h_0 = h_1 = 1 \quad (= \Delta t)$

$b_0 = \frac{1}{h_0}(y_1 - y_0) = 1$

$b_1 = \frac{1}{h_1}(y_2 - y_1) = -3$

$u_1 = 2(h_0 + h_1) = 4$

$v_1 = 6(b_1 - b_0) = -24$

$z_0 = 0$

$z_0 + 4z_1 + z_2 = -24$

$z_2 = 0$

$z_0 = z_2 = 0, \; z_1 = -6.$

If you plug into the formula for S_0, S_1, then get

$$S(x) = \begin{cases} S_0(x) = -(x+1)^3 + 3(x+1) - x & \text{for } -1 \le x \le 0 \\ S_1(x) = -(1-x^3) - x + 3(1-x) & \text{for } 0 \le x \le 1. \end{cases}$$

Check:

$S(-1) = 1$

$S(1) = -1$

$S_0(0) = -1+3 = 2$

$S_1(0) = -1+3 = 2.$

Also should check at our only interior node

$S_0'(0) = S_1'(0).$

$S_0'(x) = -3(x+1)^3 + 3 - 1 \Rightarrow$

$\Rightarrow S_0'(0) = -3+3-1 = -1.$

$S_1'(x) = 3(x+1)^3 - 1 - 3 \Rightarrow$

$S_1'(0) = 3 - 1 - 3 = -1.$

So $S_0'(0) = S_1'(0).$

Also check $S_0''(0) = S_1''(0)$:

$S_0''(x) = -6(x+1) \Rightarrow S_0''(0) = -6$

$S_1''(x) = -6(x-1) \Rightarrow S_1''(0) = -6.$

Derivatives not change too quickly (implicitly verified by taking cubic spline)