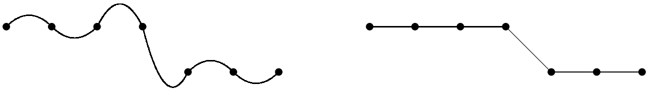
# Numerical Analysis

Raphael Hauser
with thanks to Endre Süli

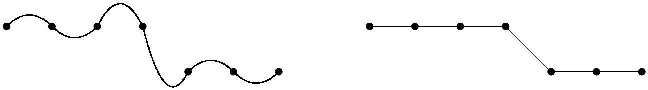
Oxford Mathematical Institute

HT 2019

# Recap on Splines

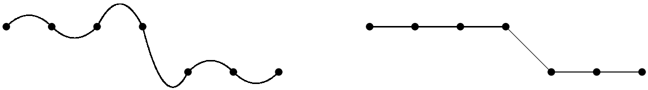


# Recap on Splines



To avoid the oscillating behaviour of interpolation by (globally defined) Lagrange interpolation polynomials, we want to construct local models that are patched together to form one of the following:

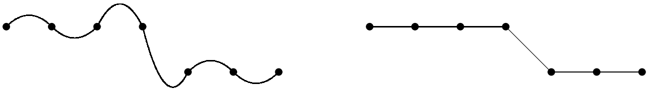
# Recap on Splines



To avoid the oscillating behaviour of interpolation by (globally defined) Lagrange interpolation polynomials, we want to construct local models that are patched together to form one of the following:

- $C^0$ function (linear splines, shown here),

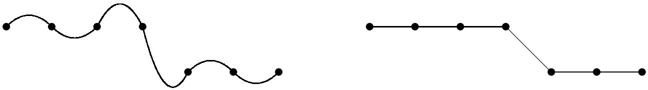
# Recap on Splines



To avoid the oscillating behaviour of interpolation by (globally defined) Lagrange interpolation polynomials, we want to construct local models that are patched together to form one of the following:

- $C^0$ function (linear splines, shown here),
- $C^1$ function (quadratic splines, not discussed further in this course),

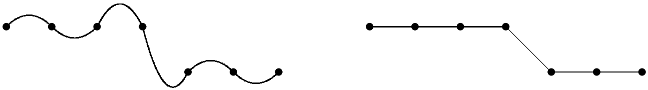
# Recap on Splines



To avoid the oscillating behaviour of interpolation by (globally defined) Lagrange interpolation polynomials, we want to construct local models that are patched together to form one of the following:

- $C^0$ function (linear splines, shown here),
- $C^1$ function (quadratic splines, not discussed further in this course),
- $C^2$ function (cubic splines, see below),

# Recap on Splines



To avoid the oscillating behaviour of interpolation by (globally defined) Lagrange interpolation polynomials, we want to construct local models that are patched together to form one of the following:

- $C^0$ function (linear splines, shown here),
- $C^1$ function (quadratic splines, not discussed further in this course),
- $C^2$ function (cubic splines, see below),
- $C^3$ ... (higher order splines, not discussed in this course).

## Theorem

*Let* $s(x)$ *be the linear spline interpolation of a function* $f \in \mathrm{C}^2[a,b]$ *at nodes* $x_0 < x_1 < \cdots < x_n$. *Then,*

$$\|f - s\|_\infty \leq \frac{1}{8} h^2 \|f''\|_\infty$$

*where* $h = \max\limits_{1 \leq i \leq n} (x_i - x_{i-1})$ *and* $\|f''\|_\infty = \max\limits_{x \in [a,b]} |f''(x)|$.

For cubic spline approximations on $a = x_0 < x_1 < \cdots < x_n = b$, we use degree 3 polynomials

For cubic spline approximations on $a = x_0 < x_1 < \cdots < x_n = b$, we use degree 3 polynomials

$$s_i(x) = \begin{cases} a_i x^3 + b_i x^2 + c_i x + d_i, & x \in (x_{i-1}, x_i), \\ 0 & \text{otherwise} \end{cases}$$

on each of the subintervals $(x_{i-1}, x_i)$, $(i = 1, \ldots, n)$,

For cubic spline approximations on $a = x_0 < x_1 < \cdots < x_n = b$, we use degree 3 polynomials

$$s_i(x) = \begin{cases} a_i x^3 + b_i x^2 + c_i x + d_i, & x \in (x_{i-1}, x_i), \\ 0 & \text{otherwise} \end{cases}$$

on each of the subintervals $(x_{i-1}, x_i)$, $(i = 1, \ldots, n)$, and we patch them together by defining

$$s(x) = \begin{cases} \sum_{i=1}^{n} s_i(x) & \text{for } x \in [a,b] \setminus \{x_0, \ldots, x_n\}, \\ f(x_i) & \text{for } x = x_i, \quad (i = 0, \ldots, n). \end{cases}$$

For cubic spline approximations on $a = x_0 < x_1 < \cdots < x_n = b$, we use degree 3 polynomials

$$s_i(x) = \begin{cases} a_i x^3 + b_i x^2 + c_i x + d_i, & x \in (x_{i-1}, x_i), \\ 0 & \text{otherwise} \end{cases}$$

on each of the subintervals $(x_{i-1}, x_i)$, $(i = 1, \ldots, n)$, and we patch them together by defining

$$s(x) = \begin{cases} \sum_{i=1}^{n} s_i(x) & \text{for } x \in [a, b] \setminus \{x_0, \ldots, x_n\}, \\ f(x_i) & \text{for } x = x_i, \quad (i = 0, \ldots, n). \end{cases}$$

To do this, we have to specify the $4n$ free parameters $a_i, b_i, c_i, d_i$.

To specify the $4n$ parameters, we can impose $4n$ conditions:

To specify the $4n$ parameters, we can impose $4n$ conditions:

(A) $s(x)$must interpolate $f$ at the nodes $x_0, \ldots, x_n$,

$$s_i(x_{i-1}^+) := \lim_{x \searrow x_{i-1}} s_i(x) = f(x_i), \quad (i = 1, \ldots, n)$$

$$s_i(x_i^-) := \lim_{x \nearrow x_i} s_i(x) = f(x_{i+1}), \quad (i = 1, \ldots, n),$$

which yields $2n$ conditions;

To specify the $4n$ parameters, we can impose $4n$ conditions:

(A) $s(x)$must interpolate $f$ at the nodes $x_0, \ldots, x_n$,

$$s_i(x_{i-1}^+) := \lim_{x \searrow x_{i-1}} s_i(x) = f(x_i), \quad (i = 1, \ldots, n)$$

$$s_i(x_i^-) := \lim_{x \nearrow x_i} s_i(x) = f(x_{i+1}), \quad (i = 1, \ldots, n),$$

which yields $2n$ conditions;

(B) first derivatives of adjacent patches must match at nodes, giving first order smoothness,

$$s_i'(x_i^-) - s_{i+1}'(x_i^+) = 0, \quad (i = 1, \ldots, n-1),$$

which yields $n-1$ conditions;

To specify the $4n$ parameters, we can impose $4n$ conditions:

(A) $s(x)$must interpolate $f$ at the nodes $x_0, \dots, x_n$,

$$s_i(x_{i-1}^+) := \lim_{x \searrow x_{i-1}} s_i(x) = f(x_i), \quad (i = 1, \dots, n)$$

$$s_i(x_i^-) := \lim_{x \nearrow x_i} s_i(x) = f(x_{i+1}), \quad (i = 1, \dots, n),$$

which yields $2n$ conditions;

(B) first derivatives of adjacent patches must match at nodes, giving first order smoothness,

$$s_i'(x_i^-) - s_{i+1}'(x_i^+) = 0, \quad (i = 1, \dots, n-1),$$

which yields $n-1$ conditions;

(C) second derivatives of adjacent patches must match at nodes, giving second order smoothness,

$$s_i''(x_i^-) - s_{i+1}''(x_i^+) = 0, \quad (i = 1, \dots, n-1),$$

which yields $n-1$ conditions.

For the final two conditions we have a choice:

For the final two conditions we have a choice:

(a) specify $s'(x_0^+) = f'(x_0)$ and $s'(x_n^-) = f'(x_n)$ — this gives a **Hermite** cubic spline ; or

For the final two conditions we have a choice:

(a) specify $s'(x_0^+) = f'(x_0)$ and $s'(x_n^-) = f'(x_n)$ — this gives a **Hermite** cubic spline ; or

(b) specify $s''(x_0^+) = 0 = s''(x_n^-)$ — this gives a **natural** cubic spline; or

For the final two conditions we have a choice:

(a) specify $s'(x_0^+) = f'(x_0)$ and $s'(x_n^-) = f'(x_n)$ — this gives a **Hermite** cubic spline ; or

(b) specify $s''(x_0^+) = 0 = s''(x_n^-)$ — this gives a **natural** cubic spline; or

(c) enforce continuity of $s'''$ at $x_1$ and $x_{n-1}$ (which implies that the first two pieces are the same cubic spline, i.e., on $[x_0, x_2]$, and similarly for the last two pieces, i.e., on $[x_{n-2}, x_n]$, from which it follows that $x_1$ and $x_{n-1}$ are not knots! — this is usually described as the '**not a knot**' end-conditions).

We may write Conditions (A), (B) and (C) complemented with one of (a), (b) or (c) in matrix form as

$$Ay = g, \tag{14.1}$$

with

$$y = (a_1, b_1, c_1, d_1, a_2, \ldots, d_{n-1}, a_n, b_n, c_n, d_n)^{\mathrm{T}}$$

and the various entries of $g$ are $f(x_i)$, $i \in 0, 1, \ldots, n$, and $f'(x_0)$, $f'(x_n)$ for (a), and zeros for (b).

We may write Conditions (A), (B) and (C) complemented with one of (a), (b) or (c) in matrix form as

$$Ay = g, \qquad (14.1)$$

with

$$y = (a_1, b_1, c_1, d_1, a_2, \ldots, d_{n-1}, a_n, b_n, c_n, d_n)^{\mathrm{T}}$$

and the various entries of $g$ are $f(x_i)$, $i \in 0, 1, \ldots, n$, and $f'(x_0)$, $f'(x_n)$ for (a), and zeros for (b).

So if $A$ is nonsingular, this implies that $y = A^{-1}g$, that is there is a unique set of coefficients $\{a_1, b_1, c_1, d_1, a_2, \ldots, d_{n-1}, a_n, b_n, c_n, d_n\}$.

We may write Conditions (A), (B) and (C) complemented with one of (a), (b) or (c) in matrix form as

$$Ay = g, \tag{14.1}$$

with

$$y = (a_1, b_1, c_1, d_1, a_2, \dots, d_{n-1}, a_n, b_n, c_n, d_n)^{\mathrm{T}}$$

and the various entries of $g$ are $f(x_i)$, $i \in 0, 1, \dots, n$, and $f'(x_0)$, $f'(x_n)$ for (a), and zeros for (b).

So if $A$ is nonsingular, this implies that $y = A^{-1}g$, that is there is a unique set of coefficients $\{a_1, b_1, c_1, d_1, a_2, \dots, d_{n-1}, a_n, b_n, c_n, d_n\}$.

We now prove that if $Ay = 0$ then $y = 0$, and thus that $A$ is nonsingular for cases (a) and (b) — it is also possible, but more complicated, to show this for case (c).

### Theorem

*If* $f(x_i) = 0$ *at the knots* $x_i$, $i = 1, \ldots, n$, *and* $f'(x_0) = 0 = f'(x_n)$ *for case (a), then* $s(x) = 0$ *for all* $x \in [x_0, x_n]$.

## Theorem

*If $f(x_i) = 0$ at the knots $x_i$, $i = 1, \ldots, n$, and $f'(x_0) = 0 = f'(x_n)$ for case (a), then $s(x) = 0$ for all $x \in [x_0, x_n]$.*

Proof. Consider

$$\begin{aligned}\int_{x_0}^{x_n} (s''(x))^2 \, \mathrm{d}x = & \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} (s_i''(x))^2 \, \mathrm{d}x \\ = & \sum_{i=1}^{n} \left[ s_i'(x) s_i''(x) \right]_{x_{i-1}}^{x_i} - \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} s_i'(x) s_i'''(x) \, \mathrm{d}x\end{aligned}$$

using integration by parts.

## Theorem

*If $f(x_i) = 0$ at the knots $x_i$, $i = 1, \ldots, n$, and $f'(x_0) = 0 = f'(x_n)$ for case (a), then $s(x) = 0$ for all $x \in [x_0, x_n]$.*

Proof. Consider

$$\begin{aligned}\int_{x_0}^{x_n} (s''(x))^2 \,\mathrm{d}x &= \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} (s_i''(x))^2 \,\mathrm{d}x \\ &= \sum_{i=1}^{n} \left[s_i'(x) s_i''(x)\right]_{x_{i-1}}^{x_i} - \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} s_i'(x) s_i'''(x) \,\mathrm{d}x\end{aligned}$$

using integration by parts. However,

$$\int_{x_{i-1}}^{x_i} s_i'(x) s_i'''(x) \,\mathrm{d}x = s_i'''(x) \int_{x_{i-1}}^{x_i} s_i'(x) \,\mathrm{d}x = s_i'''(x) \left[s_i(x)\right]_{x_{i-1}}^{x_i} = 0$$

since $s_i'''(x)$ is constant on the interval $(x_{i-1}, x_i)$ and $s_i(x_{i-1}) = 0 = s_i(x_i)$.

Thus, matching first and second derivatives at the knots, telescopic cancellation gives

$$\begin{aligned}
\int_{x_0}^{x_n} (s''(x))^2 \,\mathrm{d}x &= \sum_{i=1}^{n} \left[ s_i'(x) s_i''(x) \right]_{x_{i-1}}^{x_i} \\
&= s_1'(x_1)s_1''(x_1) - s_1'(x_0)s_1''(x_0) \\
&\quad + s_2'(x_2)s_2''(x_2) - s_2'(x_1)s_2''(x_1) + \cdots \\
&\quad + s_{n-1}'(x_{n-1})s_{n-1}''(x_{n-1}) - s_{n-1}'(x_{n-2})s_{n-1}''(x_{n-2}) \\
&\quad + s_n'(x_n)s_n''(x_n) - s_n'(x_{n-1})s_n''(x_{n-1}) \\
&= s_n'(x_n)s_n''(x_n) - s_1'(x_0)s_1''(x_0).
\end{aligned}$$

Thus, matching first and second derivatives at the knots, telescopic cancellation gives

$$\begin{aligned}
\int_{x_0}^{x_n} (s''(x))^2 \,\mathrm{d}x = & \sum_{i=1}^{n} \left[ s_i'(x) s_i''(x) \right]_{x_{i-1}}^{x_i} \\
= & \; s_1'(x_1)s_1''(x_1) - s_1'(x_0)s_1''(x_0) \\
& + s_2'(x_2)s_2''(x_2) - s_2'(x_1)s_2''(x_1) + \cdots \\
& + s_{n-1}'(x_{n-1})s_{n-1}''(x_{n-1}) - s_{n-1}'(x_{n-2})s_{n-1}''(x_{n-2}) \\
& + s_n'(x_n)s_n''(x_n) - s_n'(x_{n-1})s_n''(x_{n-1}) \\
= & \; s_n'(x_n)s_n''(x_n) - s_1'(x_0)s_1''(x_0).
\end{aligned}$$

However, in case (a), $f'(x_0) = 0 = f'(x_n) \implies s_1'(x_0) = 0 = s_n'(x_n)$, while in case (b) $s_1''(x_0) = 0 = s_n''(x_n)$.

Thus, matching first and second derivatives at the knots, telescopic cancellation gives

$$\begin{aligned}
\int_{x_0}^{x_n} (s''(x))^2 \,\mathrm{d}x &= \sum_{i=1}^{n} \left[ s_i'(x) s_i''(x) \right]_{x_{i-1}}^{x_i} \\
&= s_1'(x_1)s_1''(x_1) - s_1'(x_0)s_1''(x_0) \\
&\quad + s_2'(x_2)s_2''(x_2) - s_2'(x_1)s_2''(x_1) + \cdots \\
&\quad + s_{n-1}'(x_{n-1})s_{n-1}''(x_{n-1}) - s_{n-1}'(x_{n-2})s_{n-1}''(x_{n-2}) \\
&\quad + s_n'(x_n)s_n''(x_n) - s_n'(x_{n-1})s_n''(x_{n-1}) \\
&= s_n'(x_n)s_n''(x_n) - s_1'(x_0)s_1''(x_0).
\end{aligned}$$

However, in case (a), $f'(x_0) = 0 = f'(x_n) \implies s_1'(x_0) = 0 = s_n'(x_n)$, while in case (b) $s_1''(x_0) = 0 = s_n''(x_n)$. Thus

$$\int_{x_0}^{x_n} (s''(x))^2 \,\mathrm{d}x = 0,$$

which implies that $s_i''(x) = 0$ and thus $s_i(x) = c_i x + d_i$.

Thus, matching first and second derivatives at the knots, telescopic cancellation gives

$$\begin{aligned}
\int_{x_0}^{x_n} (s''(x))^2 \,\mathrm{d}x &= \sum_{i=1}^{n} \left[ s_i'(x) s_i''(x) \right]_{x_{i-1}}^{x_i} \\
&= s_1'(x_1)s_1''(x_1) - s_1'(x_0)s_1''(x_0) \\
&\quad + s_2'(x_2)s_2''(x_2) - s_2'(x_1)s_2''(x_1) + \cdots \\
&\quad + s_{n-1}'(x_{n-1})s_{n-1}''(x_{n-1}) - s_{n-1}'(x_{n-2})s_{n-1}''(x_{n-2}) \\
&\quad + s_n'(x_n)s_n''(x_n) - s_n'(x_{n-1})s_n''(x_{n-1}) \\
&= s_n'(x_n)s_n''(x_n) - s_1'(x_0)s_1''(x_0).
\end{aligned}$$

However, in case (a), $f'(x_0) = 0 = f'(x_n) \implies s_1'(x_0) = 0 = s_n'(x_n)$, while in case (b) $s_1''(x_0) = 0 = s_n''(x_n)$. Thus

$$\int_{x_0}^{x_n} (s''(x))^2 \,\mathrm{d}x = 0,$$

which implies that $s_i''(x) = 0$ and thus $s_i(x) = c_i x + d_i$. Since however $s(x_{i-1}) = 0 = s(x_i)$, $s(x)$ is identically zero on $[x_0, x_n]$. □

**Constructing cubic splines.** Note that (14.1) provides a constructive method for finding an interpolating spline, but generally this is not used. Motivated by the next result, it is better to find a good basis.

Constructing cubic splines. Note that (14.1) provides a constructive method for finding an interpolating spline, but generally this is not used. Motivated by the next result, it is better to find a good basis.

### Proposition

*The set of natural cubic splines on a given set of knots* $x_0 < x_1 < \cdots < x_n$ *is a vector space.*

Constructing cubic splines. Note that (14.1) provides a constructive method for finding an interpolating spline, but generally this is not used. Motivated by the next result, it is better to find a good basis.

## Proposition

*The set of natural cubic splines on a given set of knots $x_0 < x_1 < \cdots < x_n$ is a vector space.*

Proof. If $p, q \in \mathrm{C}^2[a, b]$ then $\alpha p + \beta q \in \mathrm{C}^2[a, b]$; also $p, q \in \Pi_3$ implies that $\alpha p + \beta q \in \Pi_3$ for every $\alpha$, $\beta \in \mathbb{R}$.

Constructing cubic splines. Note that (14.1) provides a constructive method for finding an interpolating spline, but generally this is not used. Motivated by the next result, it is better to find a good basis.

## Proposition

*The set of natural cubic splines on a given set of knots $x_0 < x_1 < \cdots < x_n$ is a vector space.*

Proof. If $p, q \in \mathrm{C}^2[a, b]$ then $\alpha p + \beta q \in \mathrm{C}^2[a, b]$; also $p, q \in \Pi_3$ implies that $\alpha p + \beta q \in \Pi_3$ for every $\alpha$, $\beta \in \mathbb{R}$.

Finally, the natural end-conditions (b) imply that

$$(\alpha p + \beta q)''(x_0) = 0 = (\alpha p + \beta q)''(x_n)$$

whenever $p''$ and $q''$ are zero at $x_0$ and $x_n$. $\square$

Best spline bases: the **Cardinal splines**, $C_i$, $i = 0, 1, \ldots, n$, defined as the interpolatory natural cubic splines satisfying

$$C_i(x_j) = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j, \end{cases}$$

are a basis for which

$$s(x) = \sum_{i=0}^{n} f(x_i) C_i(x)$$

is the interpolatory natural cubic spline to $f$.

Best spline bases: the **Cardinal splines**, $C_i$, $i = 0, 1, \dots, n$, defined as the interpolatory natural cubic splines satisfying

$$C_i(x_j) = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j, \end{cases}$$

are a basis for which

$$s(x) = \sum_{i=0}^{n} f(x_i) C_i(x)$$

is the interpolatory natural cubic spline to $f$.

Preferred are the **B-splines** (locally) defined by

$$B_i(x_i) = 1 \qquad \text{for } i = 2, 3, \dots, n-2,$$

$B_i(x) \equiv 0$ for $x \notin (x_{i-2}, x_{i+2})$, $B_i$ a cubic spline with knots $x_j$, $j = 0, 1, \dots, n$, with special definitions for $B_0$, $B_1$, $B_{n-1}$ and $B_n$.

Example/construction: Cubic $B$-spline with knots $0, 1, 2, 3, 4$. On $[0, 1]$,

$$B(x) = ax^3$$

for some $a$ in order that $B$, $B'$ and $B''$ are continuous at $x = 0$ (recall that $B(x)$ is required to be identically zero for $x < 0$). So

$$B(1) = a, \quad B'(1) = 3a, \quad \text{and} \quad B''(1) = 6a.$$

Example/construction: Cubic $B$-spline with knots $0, 1, 2, 3, 4$. On $[0, 1]$,

$$B(x) = ax^3$$

for some $a$ in order that $B$, $B'$ and $B''$ are continuous at $x = 0$ (recall that $B(x)$ is required to be identically zero for $x < 0$). So

$$B(1) = a, \quad B'(1) = 3a, \quad \text{and} \quad B''(1) = 6a.$$

On $[1, 2]$, since $B$ is a cubic polynomial, using Taylor's Theorem,

$$\begin{aligned} B(x) = \quad & B(1) + B'(1)(x-1) + \frac{B''(1)}{2}(x-1)^2 + \beta(x-1)^3 \\ = \quad & a + 3a(x-1) + 3a(x-1)^2 + \beta(x-1)^3 \end{aligned}$$

for some $\beta$, and since we require $B(2) = 1$, then $\beta = 1 - 7a$.

Example/construction: Cubic $B$-spline with knots $0, 1, 2, 3, 4$. On $[0, 1]$,

$$B(x) = ax^3$$

for some $a$ in order that $B$, $B'$ and $B''$ are continuous at $x = 0$ (recall that $B(x)$ is required to be identically zero for $x < 0$). So

$$B(1) = a, \quad B'(1) = 3a, \quad \text{and} \quad B''(1) = 6a.$$

On $[1, 2]$, since $B$ is a cubic polynomial, using Taylor's Theorem,

$$\begin{aligned} B(x) = \quad & B(1) + B'(1)(x-1) + \frac{B''(1)}{2}(x-1)^2 + \beta(x-1)^3 \\ = \quad & a + 3a(x-1) + 3a(x-1)^2 + \beta(x-1)^3 \end{aligned}$$

for some $\beta$, and since we require $B(2) = 1$, then $\beta = 1 - 7a$. Now, in order to continue, by symmetry, we must have $B'(2) = 0$, i.e.,

$$3a + 6a(x-1)_{x=2} + 3(1-7a)(x-1)^2_{x=2} = 3 - 12a = 0$$

and hence $a = \frac{1}{4}$.

So

$$B(x) = \begin{cases} 0 & \text{for } x < 0 \\ \frac{1}{4}x^3 & \text{for } x \in [0,1] \\ -\frac{3}{4}(x-1)^3 + \frac{3}{4}(x-1)^2 + \frac{3}{4}(x-1) + \frac{1}{4} & \text{for } x \in [1,2] \\ -\frac{3}{4}(3-x)^3 + \frac{3}{4}(3-x)^2 + \frac{3}{4}(3-x) + \frac{1}{4} & \text{for } x \in [2,3] \\ \frac{1}{4}(4-x)^3 & \text{for } x \in [3,4] \\ 0 & \text{for } x > 4. \end{cases}$$

So

$$B(x) = \begin{cases} 0 & \text{for } x < 0 \\ \frac{1}{4}x^3 & \text{for } x \in [0,1] \\ -\frac{3}{4}(x-1)^3 + \frac{3}{4}(x-1)^2 + \frac{3}{4}(x-1) + \frac{1}{4} & \text{for } x \in [1,2] \\ -\frac{3}{4}(3-x)^3 + \frac{3}{4}(3-x)^2 + \frac{3}{4}(3-x) + \frac{1}{4} & \text{for } x \in [2,3] \\ \frac{1}{4}(4-x)^3 & \text{for } x \in [3,4] \\ 0 & \text{for } x > 4. \end{cases}$$

More generally: $B$-spline on $x_i = a + hi$, where $h = (b-a)/n$.

$$B_i(x) = \begin{cases} 0 & \text{for } x < x_{i-2} \\ \dfrac{(x-x_{i-2})^3}{4h^3} & \text{for } x \in [x_{i-2}, x_{i-1}] \\ -\dfrac{3(x-x_{i-1})^3}{4h^3} + \dfrac{3(x-x_{i-1})^2}{4h^2} + \dfrac{3(x-x_{i-1})}{4h} + \dfrac{1}{4} & \text{for } x \in [x_{i-1}, x_i] \\ -\dfrac{3(x_{i+1}-x)^3}{4h^3} + \dfrac{3(x_{i+1}-x)^2}{4h^2} + \dfrac{3(x_{i+1}-x)}{4h} + \dfrac{1}{4} & \text{for } x \in [x_i, x_{i+1}] \\ \dfrac{(x_{i+2}-x)^3}{4h^3} & \text{for } x \in [x_{i+1}, x_{i+2}] \\ 0 & \text{for } x > x_{i+2}. \end{cases}$$