Interpolation in general

Given $(n+1)$ data points $(x_0, y_0), \cdots, (x_n, y_n)$ with x_i's all distinct. (1)
Find a function $f(x) \in \Pi_n \stackrel{\text{def}}{=} \{g : g(x) \text{ is a polynomial with } \deg(g) \leq n\}$ such that f passes through all $(n+1)$ data points.

Definition 1 (1) called the *interpolation* problem and such f is called the *interpolating function* or *interpolating polynomial.*

Theorem 1 *If (1) has a solution, it must be unique.*

(*pf.*) Suppose $f, g \in \Pi_n$, $f \neq g$ both solves $(\star)$. Then $(f-g)(x_i) = 0$ for all $i = 0, \cdots, n \Rightarrow (f-g)(x)$ is a polynomial of $\deg(f-g) \geq n+1$, contradicts $f, g \in \Pi_n$. □

Theorem 2 *(1) has a solution.*

(*pf.*) Suppose $f(x) = a_0 + a_1 x + \cdots + a_n x^n \in \Pi_n$ with $\begin{cases} a_0 + a_1 x_0 + \cdots + a_n x_0^n & = y_0, \\ \vdots & \vdots \\ a_n + a_1 x_n + \cdots + a_n x_n^n & = y_n. \end{cases}$

Know x_i's all distinct, $|\det A| = \left|\prod_{i \neq j}(x_i - x_j)\right| \neq 0 \iff$ the solution $(a_0, \cdots, a_n)$, i.e. $f(x)$, exists and is unique. □

Lagrange method

$f(x) = y_0 L_0(x) + \cdots + y_n L_n(x)$, where $L_i(x) \stackrel{\text{def}}{=} \prod_{j \neq i} \frac{(x - x_j)}{(x_i - x_j)}$ for $i = 0, 1, \cdots, n$;
Apparently, $f \in \Pi_n$ and f solves $(\star)$.

Theorem 3 Π_n *is a vector space of dimension* $(n+1)$.

$\Pi_n = \langle 1, x, x^2, \cdots, x^n \rangle = \langle L_0(x), \cdots, L_n(x) \rangle = \cdots$, generated by different bases.

Newton Divided Differences

Suppose $P_{01}(x) = f_0 + f_{01}(x - x_0)$ passing through $(x_0, f_0), (x_1, f_1)$, then $f_{01} = \frac{f_0 - f_1}{x_0 - x_1}$ by point-slope formula.

Suppose $P_{12}(x) = f_1 + f_{12}(x - x_1)$ passing through $(x_1, f_1), (x_2, f_2)$, then $f_{12} = \frac{f_1 - f_2}{x_1 - x_2}$ by point-slope formula.

Suppose $P_{012}(x) = f_0 + f_{01}(x - x_0) + f_{012}(x - x_0)(x - x_1)$ passing through (x_0, f_0),(x_1, f_1),(x_2, f_2), then $\begin{cases} f_{01} = \frac{f_0 - f_1}{x_0 - x_1} \\ f_{012} = \frac{f_{01} - f_{12}}{x_0 - x_2} \end{cases}$.

The dependency of coefficents is shown below:

$$\begin{array}{c|ccccc} x_0 & f_0 \searrow & & & & \\ & & f_{01} \searrow & & & \\ x_1 & f_1 \begin{smallmatrix}\nearrow \\ \searrow\end{smallmatrix} & & f_{012} & & \\ & & f_{12} \nearrow & & & \\ x_2 & f_2 \nearrow & & & & \end{array}$$

In general, given $(n+1)$ data points $(x_0, f_0), \cdots, (x_n, f_n)$ with all x_i's distinct, then the interpolating polynomial $f(x) \in \Pi_n$ can be obtained by completing the table of *Newton divided differences*:

$$f_{i_0,\cdots,i_k} = \frac{f_{i_1,\cdots,i_k} - f_{i_0,\cdots,i_{k-1}}}{x_{i_k} - x_{i_0}}, \textit{ the } k^{th} \textit{ divided difference, and}$$

$$P_{i_0,\cdots,i_k}(x) = f_{i_0} + f_{i_0,i_1}(x - x_{i_0}) + \cdots + f_{i_0,\cdots,i_k}(x - x_{i_0}) \cdots (x - x_{i_{k-1}})$$

is the interpolating polynomial through $(x_{i_0}, f_{i_0}), \cdots, (x_{i_k}, f_{i_k})$. Newton's construction of $P_{i_0,\cdots,i_k}(x)$ leads to *Neville's Algorithm* as well:

Theorem 4 *If $P_{i_0,\cdots,i_{k-1}}(x) \in \Pi_{k-1}$ passing through $(x_{i_0}, f_{i_0}), \cdots, (x_{i_{k-1}}, f_{i_{k-1}})$, and $P_{i_1,\cdots,i_k}(x) \in \Pi_{k-1}$ passing through $(x_{i_1}, f_{i_1}), \cdots, (x_{i_k}, f_{i_k})$, then*

$$P_{i_0,\cdots,i_k}(x) = \frac{(x - x_{i_0})P_{i_1,\cdots,i_k}(x) - (x - x_{i_k})P_{i_0,\cdots,i_{k-1}}(x)}{x_{i_k} - x_{i_0}} \in \Pi_k$$

passes through $(x_{i_0}, f_{i_0}), \cdots, (x_{i_k}, f_{i_k})$.

Note $\Pi_n = \left\langle 1, (x - x_0), (x - x_0)(x - x_1), \cdots, (x - x_0) \cdots (x - x_{n-1}) \right\rangle$ now.

General Hermite Interpolation

Specify

$$\begin{cases} f(x_0), f'(x_0), \ldots, f^{(n_0-1)}(x_0), & (n_0 \geq 1) \\ \vdots & \\ f(x_m), f'(x_m), \ldots, f^{(n_m-1)}(x_m), & (n_m \geq 1) \end{cases} \tag{2}$$

with distince $\{x_i\}_{i=0}^m$ and find $f \in \Pi_n$, where $n + 1 = \sum_{i=0}^m n_i = \#$conditions.

Theorem 5 *Hermite interpolating polynomial is unique.*

(*pf.*) (Cannot apply Theorem 1 since these two are different kinds interpolations.) Let $f, g \in \Pi_n$ both solve (2). Then $(f-g)^{(i)}(x_j) = 0\,\forall\, i = 0, \cdots, n_j - 1, \Rightarrow$ $f - g$ has factors $(x-x_0)^{n_0}, \cdots, (x-x_m)^{n_m}$, since x_i's all distinct, $\deg(f-g) \geq \sum_{i=0}^{m} n_i = n+1$, a contradition. □

Theorem 6 *Solution of (2) exists.*

Question: Can we switch the order like what we do onto the Newton divided difference table?

Piecewise Interpolation

Instead of finding a interpolating polynomial defined explicitly on $\mathbb{R}$, we can loosen this restriction by defining interpolating functions piecewisely, with or without smoothness conditions.

0.1 Splines

Given data points $(x_0, y_0), \cdots, (x_m, y_m)$ with all x_i's distinct, define a corresponding interpolating function $S(x)$ satisfying the following conditions:

$$\begin{cases} S_j(x) \overset{\text{def}}{=} S(x)\Big|_{[x_j, x_{j+1}]} \in \Pi_n \ \forall j = 0, \cdots, m-1; \\ S^{(0)}(x), \cdots, S^{(n-1)}(x) \text{ all continuous;} \end{cases} \tag{3}$$

Such an $S(x)$ is called an *n-spline*. Let's take a look at $n = 3$ *cubic spline* case:

$$\begin{cases} S_j(x) \overset{\text{def}}{=} S(x)\Big|_{[x_j, x_{j+1}]} \in \Pi_3 \ \forall j = 0, \cdots, m-1; \\ S(x), S'(x), S''(x) \text{ all continuous;} \end{cases} \tag{4}$$

Let $\begin{array}{c} h_j := x_{j+1} - x_j, \\ j = 0, \cdots, m-1, \end{array}$ and $\begin{array}{c} M_j := S''(x_j), \\ j = 0, \cdots, m. \end{array}$ Then

$$S_j''(x) = M_j + \frac{M_{j+1} - M_j}{x_{j+1} - x_j}(x - x_j) = M_j + \frac{M_{j+1} - M_j}{h_j}(x - x_j) \tag{5}$$

by point-slope formula, $\Rightarrow$ the continuity of S'' is automatically satisfied.

By the continuity of S, we can derive $S_j(x)$ formula in the form

$$a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$$

in terms of M_* and h_*. Integrating (5) twice:

$$S_j(x) = a_j + b_j(x - x_j) + \underbrace{\tfrac{M_j}{2}}_{c_j}(x - x_j)^2 + \underbrace{\tfrac{M_{j+1} - M_j}{6h_j}}_{d_j}(x - x_j)^3 \tag{6}$$

Directly from (6):

$$S_j(x_j) = y_j, \Rightarrow a_j = y_j,$$
$$S_j(x_{j+1}) = y_{j+1}, \Rightarrow y_{j+1} = y_j + b_j \cdot h_j + \frac{M_j}{2}h_j^2 + \frac{M_{j+1}-M_j}{6}h_j^2,$$
$$b_j = \left((y_{j+1}-y_j) - (\frac{2M_j+M_{j+1}}{6})h_j^2\right)/h_j = \frac{y_{j+1}-y_j}{h_j} - \frac{2M_j+M_{j+1}}{6}h_j.$$

By the continuity of S', we can derive a relationship among $M_{j-1}, M_j,$ and M_{j+1}. Differentiate (6):

$$S_j'(x) = (\frac{y_{j+1}-y_j}{h_j}) - (\frac{2M_j+M_{j+1}}{6})h_j + M_j(x-x_j) + (\frac{M_{j+1}-M_j}{2h_j})(x-x_j)^2. \qquad (7)$$

$$S'(x_j^-) = S'(x_j^+), \text{ i.e. } S_{j-1}'(x_j) = S_j'(x_j), \; j = 1, \cdots, m-1, \text{ if and only if}$$

$$\begin{array}{c}(\frac{h_{j-1}}{6})M_{j-1} + (\frac{h_{j-1}+h_j}{6})M_j + (\frac{h_j}{6})M_{j+1} = (\frac{y_{j+1}-y_j}{h_j}) - (\frac{y_j-y_{j-1}}{h_{j-1}}), \\ j = 1, \cdots, m-1\end{array} \qquad (8)$$

(8), plus two additional boundary conditions (eg. free B.C.: $S''(x_0) = S''(x_m) = 0$, or, clamped B.C.), becomes a square system in M_* ($m+1$ eqn's and $m+1$ M_j's). $S_j(x)$ (6) is obtained after a_j, b_j, M_j solved.

Bézier Curves

A parameterized curve $\mathbf{r}(t) = (x(t), y(t)),\; t \in [0,1]$ with $\begin{cases}\mathbf{r}(0) = (x_0, y_0) \\ \dot{\mathbf{r}}(0) = (u_0, v_0)\end{cases}$ and $\begin{cases}\mathbf{r}(1) = (x_1, y_1) \\ \dot{\mathbf{r}}(1) = (u_1, v_1)\end{cases}$ specified. If $x, y \in \Pi_3$, then we can uniquely determine $\mathbf{r}(t)$:

For $x(t)$ cubic interpolation (by Hermite's interpolation),

t	0^{th}	1^{st}	2^{nd}	3^{rd}
0	x_0			
		u_0		
0	x_0		$\overbrace{x_1-x_0-u_0}^{b}$	
		x_1-x_0		$\overbrace{u_0+u_1-2(x_1-x_0)}^{a}$
1	x_1		$u_1-x_1+x_0$	
		u_1		
1	x_1			

Then

$$\begin{aligned}x(t) &= x_0 + u_0t + bt^2 + at^2(t-1) \\ &= x_0 + u_0t + (b-a)t^2 + at^3 \\ &= x_0 + u_0t + \big(3(x_1-x_0)-2u_0-u_1\big)t^2 + \big(u_0+u_1-2(x_1-x_0)\big)t^3 .\end{aligned}$$

Similarly,

$$y(t) = y_0 + v_0t + \big(3(y_1-y_0)-2v_0-v_1\big)t^2 + \big(v_0+v_1-2(y_1-y_0)\big)t^3 .$$

A curve determined by that is called a *Bézier curve*.

Numerical Integration(quadrature about $\int_a^b f(x)\,dx$)

Trapezoidal Rule(2 point interpolation) Let $h := x_0 - x_1$,

$$f(x) = \underbrace{\tfrac{(x-x_1)}{(x_0-x_1)} f(x_0) + \tfrac{(x-x_0)}{(x_1-x_0)} f(x_1)}_{\text{interpolating polynomial}} + \underbrace{(x-x_0)(x-x_1)\tfrac{f''(\xi_1)}{2!}}_{\text{error term}}$$

for some $\xi_1 \in [x_0, x_1]$, then

$$\int_{x_0}^{x_1} f(x)\,dx = \tfrac{h}{2}(f(x_0) + f(x_1)) - \tfrac{h^3}{12} f''(\bar{\xi}_1)$$

for some $\bar{\xi}_1 \in [x_0, x_1]$.

Composite Trapezoidal Rule: divide $[a, b]$ into n equal pieces $a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b$, $h := \frac{b-a}{n}$, then

$$\int_a^b f(x)\,dx = \tfrac{h}{2}\Big\{ f(x_0) + 2f(x_1) + \cdots + 2f(x_{n-1}) + f(x_n) \Big\} - \tfrac{b-a}{12} f''(\bar{\xi}) \cdot h^2$$

for some $\bar{\xi} \in [a, b]$. If f'' is continuous on $[a, b]$, then f is integrable on $[a, b]$ and the above converges as $h \to 0^+$.

Notice that **composite trapezoidal rule is an order 2 method**.

Simpson's Rule(3 point interpolation) Let $h := x_{i+1} - x_i$,

$$f(x) = \underbrace{\tfrac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} f(x_0) + \tfrac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} f(x_1) + \tfrac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} f(x_2)}_{\text{interpolating polynomial}} + \underbrace{(x-x_0)(x-x_1)(x-x_2)\tfrac{f'''(\xi_2)}{3!}}_{\text{error term}}$$

for some $\xi_2 \in [x_0, x_2]$, then

$$\int_{x_0}^{x_2} f(x)\,dx = \tfrac{h}{3}(f(x_0) + 4f(x_1) + f(x_2)) - \tfrac{h^5}{90} f^{(4)}(\bar{\xi}_2)$$

for some $\bar{\xi}_2 \in [x_0, x_2]$.

Composite Simpson's Rule: divide $[a, b]$ into $2n$ equal pieces $a = x_0 < x_1 < \cdots < x_{2n-1} < x_{2n} = b$, $h := \frac{b-a}{2n}$, then

$$\int_a^b f(x)\,dx = \tfrac{h}{3}\Big\{ f(x_0) + 4f(x_1) + 2f(x_2) + \cdots + 4f(x_{2n-2}) + 2f(x_{2n-1}) + f(x_n) \Big\} - \tfrac{b-a}{90} f^{(4)}(\bar{\xi}) \cdot h^4$$

for some $\bar{\xi} \in [a, b]$. If $f^{(4)}$ is bounded on $[a, b]$, then f is integrable on $[a, b]$ and the above converges as $h \to 0^+$.

Notice that **composite Simpson's rule is an order 4 method**.

Richardson Extrapolation

Let $F(0) = \lim_{h\to 0} F(h)$ and $F(h) = F(0) + Kh^p + \mathcal{O}(h^r)\,(r > p)$ be an order p method which approximates $F(0)$. *Richardson Extrapolation* is a way to derive a new method F^* of higher precision order:

$$F^*(h) := F(h) - \frac{F(qh) - F(h)}{q^p - 1}$$

is an order r method, i.e. $F^*(h) = F(0) + \mathcal{O}(h^r)$.

Corollary 1 If $F(h) = F(0)+K_1h^{p_1}+K_2h^{p_2}\cdots+K_nh^{p_n}+\cdots$, $0 < p_1 < p_2 < \cdots$, define $F_1(h) := F(h)$, $F_2(h) := F_1^*(h)$, $F_3 := F_2^*, \cdots$, then for $j \geq 1$,

$$F_{j+1}(h) := F_j^*(h) = F_j(h) - \frac{F_j(qh) - F_j(h)}{q^{p_j} - 1} \qquad (\star)$$

is of order p_{j+1}.

(*pf.*) (Induction) $(\star)$ is true for $j = 1$. Suppose $(\star)$ is also true for $j = n - 1$, i.e. $F_n(h) = F(0) + \mathcal{O}(h^{p_n}) = F(0) + a_nh^{p_n} + a_{n+1}h^{p_{n+1}} + \cdots$, then

$$\begin{aligned} F_{n+1}(h) := {} & F_n(h) - \frac{F_n(qh) - F_n(h)}{q^{p_n} - 1} \\ = {} & F(0) + \underbrace{b_n}_{\overset{\|}{0?}} h^{p_n} + b_{n+1}h^{p_{n+1}} + \cdots \end{aligned}$$

Look at h^{p_n} term: $a_nh^{p_n} - \dfrac{a_nh^{p_n}q^{p_n} - a_nh^{p_n}}{q^{p_n} - 1} = h^{p_n}(a_n - a_n\dfrac{q^{p_n} - 1}{q^{p_n} - 1}) = 0$,

$\therefore b_n = 0$, i.e. F_{n+1} is of order p_{n+1}. □

Let $F_1(h_i) := R_{i,1}$, $R_{i,j} := F_j(h_i)$, and $h_i = \frac{h_{i-1}}{q}$. Then $R_{i,j}$ is a new approximation method improved from $R_{i,j-1}$, i.e. $\begin{matrix} R_{i,j} = R_{i,j-1}{}^* \\ \| \quad\;\; \| \\ F_j(h_i) = F_{j-1}{}^*(h_i) \end{matrix}$, and $R_{i,j}$ formula can be written in terms of $R_{*,j-1}$:

$$\begin{aligned} R_{i,j} &= F_{j-1}(h_i) - \frac{F_{j-1}(qh_i) - F_{j-1}(h_i)}{q^{p_{j-1}} - 1} \\ &= R_{i,j-1} - \frac{R_{i-1,j-1} - R_{i,j-1}}{q^{p_{j-1}} - 1} \end{aligned}$$

i.e. the dependency of $R_{i,j}$ is shown in the following table:

$$\begin{array}{llll} R_{1,1} \searrow & & & \\ & R_{2,2} \searrow & & \\ R_{2,1} \begin{matrix}\nearrow \\ \searrow\end{matrix} & & R_{3,3} \searrow & \\ & R_{3,2} \nearrow & \vdots & \ddots \\ R_{3,1} \nearrow & \vdots & & \\ \vdots & & & \end{array}$$

Romberg Integration

If $F_1(h_i) := R_{i,1}$ is the quadrature of $\int_a^b f(x)\,dx$ by means of *composite trapezoidal rule* with $[a,b]$ being sub-divided into 2^{i-1} equal pieces, i.e. $\begin{cases} q = 2\,, \\ h_i := \frac{b-a}{q^{i-1}}, \end{cases}$ Then, by *Euler-Maclaurin expansion formula*, we can show that $R_{i,1} = F_1(h_i) = F_1(0) + K_1 h_i^2 + K_2 h_i^4 + \cdots + K_n h_i^{2n} + \cdots$, i.e. $p_j = 2j$, and the above $R_{i,j}$ formula becomes

$$R_{i,j} = R_{i,j-1} - \frac{R_{i-1,j-1} - R_{i,j-1}}{4^{j-1} - 1}$$

To complete $R_{i,j}$ table, we need to compute the first column $R_{*,1}$, and $R_{i,1}$ can be derived in terms of $R_{i-1,1}$ and f:

$$\begin{aligned}
R_{i,1} &= \sum_{j=0}^{2^{i-1}-1} \frac{f(a+jh_i) + f\big(a+(j+1)h_i\big)}{2} h_i = \frac{h_i}{2}\left(f(a) + f(b) + 2\sum_{j=1}^{2^{i-1}-1} f(a+jh_i) \right) \\
&= \tfrac{1}{2}\tfrac{h_{i-1}}{2}\left(f(a) + f(b) + 2\sum_{\substack{j=1 \\ j \text{ is even}}}^{2^{i-1}-1} f(a+jh_i) + 2\sum_{\substack{j=1 \\ j \text{ is odd}}}^{2^{i-1}-1} f(a+jh_i) \right) \\
&= \tfrac{1}{2}\left[R_{i-1,1} + h_{i-1}\sum_{\substack{j=1 \\ j \text{ is odd}}}^{2^{i-1}-1} f(a+jh_i) \right] = \tfrac{1}{2}\left[R_{i-1,1} + h_{i-1}\sum_{k=1}^{2^{i-2}} f\big(a+(2k-1)h_i\big) \right] \\
&= \tfrac{1}{2}\left[R_{i-1,1} + h_{i-1}\sum_{k=1}^{2^{i-2}} f\big(a+(k-0.5)h_{i-1}\big) \right]
\end{aligned}$$

The so-called **Romberg integration** is the $j = 3$ (order 6 method) quadrature $R_{i,3}$'s

Trapzodial	Simpson	Romberg	
$R_{1,1} \searrow$			
	$R_{2,2} \searrow$		
$R_{2,1} \nearrow\searrow$		$R_{3,3} \searrow$	
	$R_{3,2} \nearrow$	$\vdots$	$\ddots$
$R_{3,1} \nearrow$	$\vdots$		
$\vdots$			

Numerical Differentiation

Given $n+1$ data points $(x_0, f(x_0)), \cdots, (x_n, f(x_n))$ (with $x_0, \cdots, x_n$ all distinct) from $f(x)$ and if Lagrange interpolation is taken:

$$f(x) = \underbrace{\sum_{i=0}^{n} f(x_i) L_{n,i}(x)}_{\text{interpolating polynomial } p(x)} + \underbrace{\tfrac{(x-x_0)\cdots(x-x_n)}{(n+1)!} f^{(n+1)}(\xi(x))}_{\text{error } \epsilon(x)},$$

where $L_{n,i}(x) := \prod_{\substack{j \neq i \\ j=0}}^{n} \frac{(x-x_j)}{(x_i-x_j)}$, Lagrange polynomials.

$$f'(x) = \underbrace{\sum_{i=0}^{n} f(x_i) L'_{n,i}(x)}_{p'(x)} + \underbrace{\tfrac{d}{dx}[\tfrac{(x-x_0)\cdots(x-x_n)}{(n+1)!}] f^{(n+1)}(\xi(x)) + \tfrac{(x-x_0)\cdots(x-x_n)}{(n+1)!} \tfrac{d}{dx}[f^{(n+1)}(\xi(x))]}_{\epsilon'(x)},$$

If $f^{(n+1)}$ is bounded and $x_0, \cdots, x_n$ are "pretty close", then $\begin{cases} f(x) \approx p(x), \\ f'(x) \approx p'(x). \end{cases}$

In particular, if $x := x_j$, then $f'(x_j) = \sum_{i=0}^{n} f(x_i) L'_{n,i}(x_j) + \prod_{\substack{i \neq j \\ i=0}}^{n} (x_j - x_i) \cdot \frac{f^{(n+1)}(\xi(x_j))}{(n+1)!}$,

$$\text{i.e. } f'(x_j) \approx p'(x_j) = \sum_{i=0}^{n} f(x_i) L'_{n,i}(x_j) :$$

$n = 1$ (two-point formula) $f'(x_0) = f'(x_1) \approx \frac{f(x_0)-f(x_1)}{x_0-x_1}$, i.e. slope of secant line.

$n = 2$ (three-point formula) $f'(x_j) \approx f(x_0)\frac{2x_j-x_1-x_2}{(x_0-x_1)(x_0-x_2)} + f(x_1)\frac{2x_j-x_0-x_2}{(x_1-x_0)(x_1-x_2)} + f(x_2)\frac{2x_j-x_0-x_1}{(x_2-x_0)(x_2-x_1)}$

Let $\begin{cases} x_1 := x_0 - h, \\ x_2 := x_0 + h, \end{cases}$ then $f'(x_0) \approx \frac{f(x_0+h)-f(x_0-h)}{2h}$, $\xrightarrow{h \to 0} f'(x_0)$.

$n = 4$ (five-point formula) Let $\begin{cases} x_1 := x_0 - h, \;\; x_2 := x_0 + h, \\ x_3 := x_0 - 2h, \; x_4 := x_0 + 2h, \end{cases}$ then

$$f'(x_0) \approx \sum_{i=0}^{n} f(x_i) L'_{n,i}(x_0) = \tfrac{4}{3}\tfrac{f(x_0+h)-f(x_0-h)}{2h} - \tfrac{1}{3}\tfrac{f(x_0+2h)-f(x_0-2h)}{4h}.$$