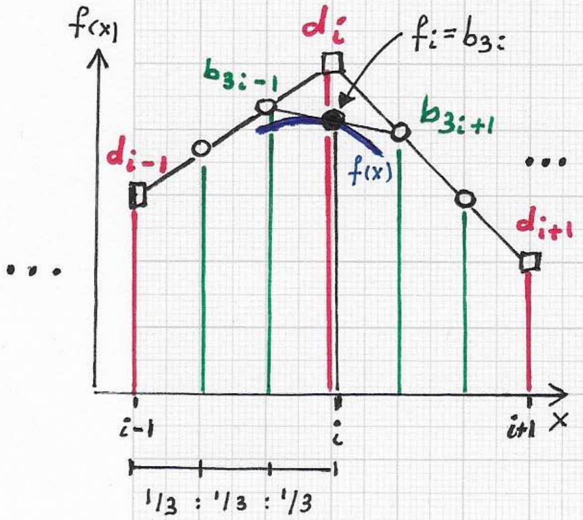
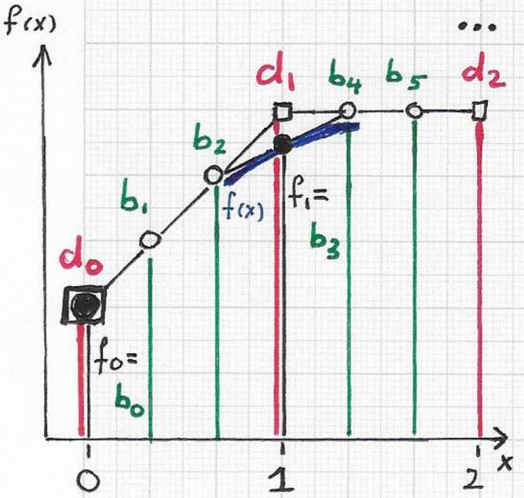
■ FRACTIONAL CALCULUS AND FEATURES – Cont'd.

• Fractional derivatives:



C^2-continuity at $x=i$



Natural end condition at $x=0$

$\Rightarrow$ Bernstein-Bézier coeffs.:

$$\boxed{\begin{aligned} b_{3i} &= f_i \ , \quad i=0\ldots n \\ b_{3i+1} &= \tfrac{2}{3}\, d_i + \tfrac{1}{3}\, d_{i+1}, \\ b_{3i+2} &= \tfrac{1}{3}\, d_i + \tfrac{2}{3}\, d_{i+1}, \\ & \qquad i=0\ldots(n-1) \end{aligned}}$$

In case one is interested in using a low-degree spline with C^2-continuity to interpolate the $(n+1)$ integer-order derivatives, the cubic spline with natural end conditions offers a viable alternative to the C^0-continuous linear spline. The construction of a natural cubic spline is briefly summarized for the problem of interpolating the values $f_0, f_1, \ldots, f_n$ given at integer-value x-coordinates $0, \ldots, n$ (see illustrations, left). When representing each cubic spline segment in Bernstein-Bézier form and enforcing C^2-continuity conditions at shared end points $(i, f_i)^T$, $i=1\ldots(n-1)$, one obtains $(n-1)$ linear equations for the so-called "de Boor control points" (coefficients) d_i:

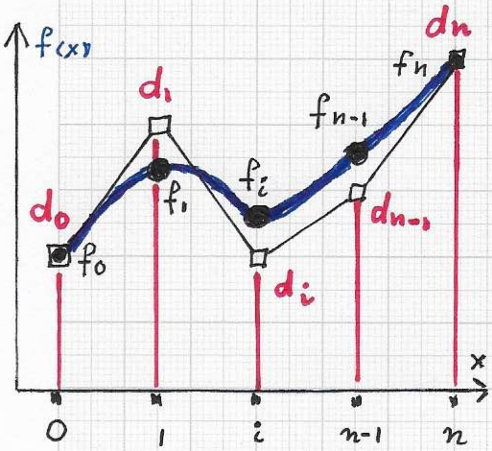
$$\boxed{d_{i-1} + 4\, d_i + d_{i+1} = 6\, f_i \ , \quad i=1\ldots(n-1)}.$$

Natural end condition for $x=0$ and $x=n$ require that $f''(0)=0$ and $f''(n)=0$, defining the needed two additional equations

$$\boxed{d_0 = f_0} \quad \text{and} \quad \boxed{d_n = f_n}.$$

The $(n+1)$ linear equations define $\{d_i\}_{i=0}^{n}$.

FRACTIONAL CALCULUS AND FEATURES – Cont'd.

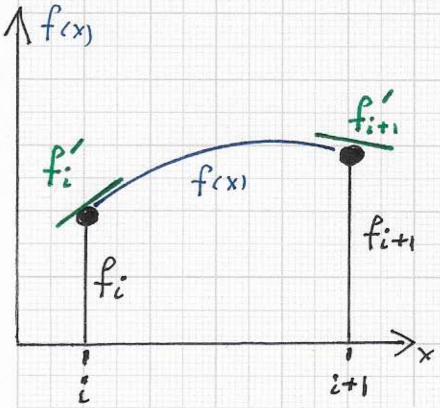


Natural cubic spline $f(x)$ over integer partition $0, 1, \ldots, n$ with defining control points/coefficients d_i for given values f_i, $i = 0 \ldots n$

• Fractional derivatives: Since the natural end conditions for the cubic spline require that $d_0 = f_0$ and $d_n = f_n$, one can determine the interior de Boor control points/coefficients by solving – in linear time – the tridiagonal linear system

$$\begin{bmatrix} 4 & 1 & & & & & \\ 1 & 4 & 1 & & & 0 & \\ & 1 & 4 & 1 & & & \\ & & & \ddots & & & \\ & & & 1 & 4 & 1 & \\ & 0 & & & 1 & 4 & 1 \\ & & & & & 1 & 4 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_{n-3} \\ d_{n-2} \\ d_{n-1} \end{bmatrix} = 6 \begin{bmatrix} f_1 - \frac{1}{6} f_0 \\ f_2 \\ f_3 \\ \vdots \\ f_{n-3} \\ f_{n-2} \\ f_{n-1} - \frac{1}{6} f_n \end{bmatrix}.$$

• Note: Concerning the possibility of computing the degree-n polynomial interpolating $(n+1)$ values $f_0, \ldots, f_n$, this possibility is not desirable as the resulting polynomial can exhibit undesirable under- and overshoot behavior.



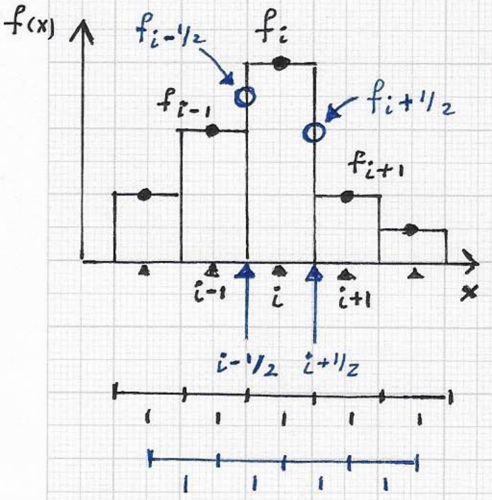
Cubic segment over $[i, i+1]$ of a C^1-continuous cubic spline defined by function and first derivative values f_i, f_{i+1}, f'_i and f'_{i+1}

• Note: The "shape" of the natural cubic spline is globally affected by each value f_i. A C^1-continuous cubic spline (Left) ensures that a value f_i and specified first derivative value f'_i affect only the spline segments over $[i-1, i]$ and $[i, i+1]$.

FRACTIONAL CALCULUS AND FEATURES — Cont'd.

• Fractional derivatives:

In the case of 2D/3D digital image/scan processing one must use FINITE DIFFERENCE FORMULAS to estimate pixel-/voxel-specific polynomial derivatives. Polynomial-based methods for derivative estimation use the principle of fitting a (relatively low-degree) polynomial to the given discrete image/scan values in a pixel's/voxel's local neighborhood. When differentiating the resulting polynomial at the location of a pixel/voxel one obtains the finite difference formula(s) in terms of a "weighted sum(s) of local function values", defining the needed derivative(s). We summarize how to derive these finite difference formulas:



Indexing scheme and equidistant, uniform spacing used to determine simple finite difference formulas

0) $f_i^{(0)} = f_i$; $f_{i+1/2}^{(0)} = f_{i+1/2} = \frac{f_i^{(0)} + f_{i+1}^{(0)}}{2} = \frac{f_i + f_{i+1}}{2}$

1) $f_i^{(1)} = f_{i+1/2}^{(0)} - f_{i-1/2}^{(0)} = (f_i + f_{i+1} - f_i - f_{i-1})/2$
$= \frac{f_{i+1} - f_{i-1}}{2}$

$f_{i+1/2}^{(1)} = f_{i+1}^{(0)} - f_i^{(0)} = f_{i+1} - f_i$

2) $f_i^{(2)} = f_{i+1/2}^{(1)} - f_{i-1/2}^{(1)} = f_{i+1} - f_i - f_i + f_{i-1}$
$= f_{i+1} - 2f_i + f_{i-1}$

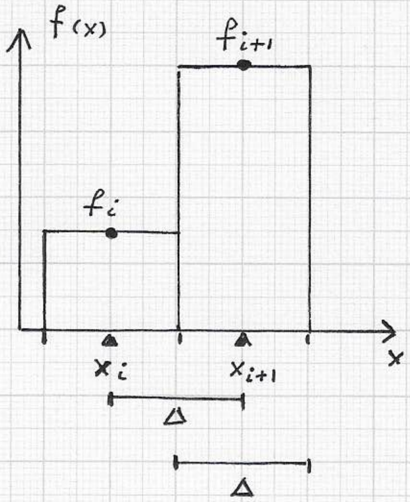
Integer-order derivative estimates $f_i^{(0)}, f_i^{(1)}, \ldots, f_i^{(K)}$ will be used to compute non-integer-order derivative estimates $f_i^{(d)}$, $0<d<K$.

FRACTIONAL CALCULUS AND FEATURES – Cont'd.

- Fractional derivatives:

2) $f^{(2)}_{i+1/2} = f^{(1)}_{i+1} - f^{(1)}_{i} = (f_{i+2} - f_i - f_{i+1} + f_{i-1})/2$

$$= \frac{f_{i+2} - f_{i+1} - f_i + f_{i-1}}{2}$$



General spacing Δ between function values

3) $f^{(3)}_{i} = f^{(2)}_{i+1/2} - f^{(2)}_{i-1/2} = (f_{i+2} - f_{i+1} - f_i + f_{i-1}$

$\quad - f_{i+1} + f_i + f_{i-1} - f_{i-2})/2$

$$= \frac{f_{i+2} - 2f_{i+1} + 2f_{i-1} - f_{i-2}}{2}$$

$f^{(3)}_{i+1/2} = f^{(2)}_{i+1} - f^{(2)}_{i} = f_{i+2} - 2f_{i+1} + f_i - f_{i+1} + 2f_i - f_{i-1}$

$\quad = f_{i+2} - 3f_{i+1} + 3f_i - f_{i-1}$

4) $f^{(4)}_{i} = f^{(3)}_{i+1/2} - f^{(3)}_{i-1/2} = f_{i+2} - 3f_{i+1} + 3f_i - f_{i-1}$

$\quad - f_{i+1} + 3f_i - 3f_{i-1} + f_{i-2}$

$\quad = f_{i+2} - 4f_{i+1} + 6f_i - 4f_{i-1} + f_{i-2}$

$f^{(4)}_{i+1/2} = f^{(3)}_{i+1} - f^{(3)}_{i} = (f_{i+3} - 2f_{i+2} + 2f_i - f_{i-1}$

$\quad - f_{i+2} + 2f_{i+1} - 2f_{i-1} + f_{i-2})/2$

$$= \frac{f_{i+3} - 3f_{i+2} + 2f_{i+1} + 2f_i - 3f_{i-1} + f_{i-2}}{2}$$

5) $f^{(5)}_{i} = f^{(4)}_{i+1/2} - f^{(4)}_{i-1/2} = (f_{i+3} - 3f_{i+2} + 2f_{i+1} + 2f_i - 3f_{i-1} + f_{i-2}$

$\quad - f_{i+2} + 3f_{i+1} - 2f_i - 2f_{i-1} + 3f_{i-2} - f_{i-3})/2$

$$= \frac{f_{i+3} - 4f_{i+2} + 5f_{i+1} - 5f_{i-1} + 4f_{i-2} - f_{i-3}}{2}$$

	$i-3$	$i-2$	$i-1$	i	$i+1$	$i+2$	$i+3$
$f_i^{(0)}$				1			
$f_i^{(1)}$			$-\frac{1}{2}$	0	$\frac{1}{2}$		
$f_i^{(2)}$			1	-2	1		
$f_i^{(3)}$		$-\frac{1}{2}$	1	0	-1	$\frac{1}{2}$	
$f_i^{(4)}$		1	-4	6	-4	1	
$f_i^{(5)}$	$-\frac{1}{2}$	2	$-\frac{5}{2}$	0	$\frac{5}{2}$	-2	$\frac{1}{2}$

Templates/filter masks for central difference computations used for estimating derivatives $f_i^{(0)}, \dots, f_i^{(5)}$ via local polynomial interpolation with spacing $\Delta = 1$

...

When using a more general spacing Δ between function values, the CENTRAL DIFFERENCE values ${}^{\Delta}f_i^{(j)}$ become

$${}^{\Delta}f_i^{(j)} = f_i^{(j)} / \Delta^{j} .$$

FRACTIONAL CALCULUS AND FEATURES – Cont'd.

• **Fractional derivatives:** As we are concerned with the definition of fractional derivatives for 2D images / 3D scans, we need to employ the finite difference formulas for integer-order derivative estimates for the estimation of partial derivatives $f_{i,j}^{(p,q)}$ at a pixel (i,j) or $f_{i,j,k}^{(p,q,r)}$ at a voxel (i,j,k). The integers p, q and r refer to the p^{th}, q^{th} and r^{th} derivative operator applied in i-, j- and k-direction, respectively. The differencing operators used in the 1D case are simply generalized to the 2D and 3D cases by viewing 2D images / 3D scans as tensor product generalizations of the 1D case. We provide a few low-order partial derivative formulas for the uniform, equidistant spacing $\Delta = 1$:

$f_{i,j}^{(0,0)}$:

$$\begin{array}{c|c|}
\cline{2-2} j & 1 \\
\cline{2-2} \multicolumn{1}{c}{} & \multicolumn{1}{c}{i}
\end{array}$$

$f_{i,j}^{(1,0)}$:

$$\begin{array}{c|c|c|c|}
\cline{2-4} j & -\frac{1}{2} & 0 & \frac{1}{2} \\
\cline{2-4} \multicolumn{1}{c}{} & \multicolumn{1}{c}{i-1} & \multicolumn{1}{c}{i} & \multicolumn{1}{c}{i+1}
\end{array}$$

$f_{i,j}^{(0,1)}$:

$$\begin{array}{c|c|}
\cline{2-2} j+1 & \frac{1}{2} \\
\cline{2-2} j & 0 \\
\cline{2-2} j-1 & -\frac{1}{2} \\
\cline{2-2} \multicolumn{1}{c}{} & \multicolumn{1}{c}{i}
\end{array}$$

$f_{i,j}^{(1,1)}$:

$$\begin{array}{c|c|c|c|}
\cline{2-4} j+1 & -\frac{1}{4} & 0 & \frac{1}{4} \\
\cline{2-4} j & 0 & 0 & 0 \\
\cline{2-4} j-1 & \frac{1}{4} & 0 & -\frac{1}{4} \\
\cline{2-4} \multicolumn{1}{c}{} & \multicolumn{1}{c}{i-1} & \multicolumn{1}{c}{i} & \multicolumn{1}{c}{i+1}
\end{array}$$

Local stencils and weights used to estimate partial derivatives $f_{i,j}^{(p,q)}$ at pixel (i,j)

0) $f_{i,j}^{(0,0)} = f_{i,j}$

1) $f_{i,j}^{(1,0)} = \left(f_{i+1,j}^{(0,0)} - f_{i-1,j}^{(0,0)}\right)/2 = \dfrac{f_{i+1,j} - f_{i-1,j}}{2}$

$f_{i,j}^{(0,1)} = \left(f_{i,j+1}^{(0,0)} - f_{i,j-1}^{(0,0)}\right)/2 = \dfrac{f_{i,j+1} - f_{i,j-1}}{2}$

$$\begin{aligned}
f_{i,j}^{(1,1)} &= \left(f_{i,j+1}^{(1,0)} - f_{i,j-1}^{(1,0)}\right)/2 \\
&= \Big(\left(f_{i+1,j+1} - f_{i-1,j+1}\right)/2 - \left(f_{i+1,j-1} - f_{i-1,j-1}\right)/2\Big)/2 \\
&= \frac{f_{i+1,j+1} - f_{i-1,j+1} - f_{i+1,j-1} + f_{i-1,j-1}}{4}
\end{aligned}$$