Lagrange Polynomial

- Useful when the grid points are fixed but function values may be changing (estimating the temperature at a point using the measured temperatures at nearby points)
- The value of the Lagrange polynomials at the desired point need to be calculated only once
- Then, we just need to multiply these values with the corresponding temperatures
- What if a new measurement is added?
- The polynomials will need to be recomputed

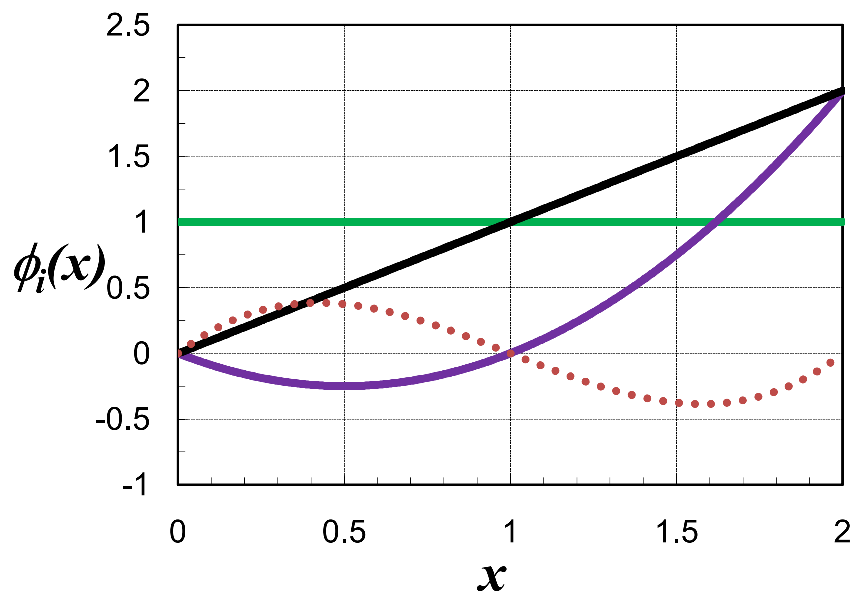
Interpolation: Newton's divided difference

$$f_n(x) = \sum_{j=0}^{n} c_j \phi_j(x)$$

- The basis function, ϕ_i, is an i^{th}-degree polynomial, which is zero at all "previous" points. ϕ_0=1, and, for i>0,

$$\phi_i(x) = \prod_{j=0}^{i-1} (x - x_j)$$

- For example, using the same data (x=0,1,2):



Newton's divided difference

$$f_n(x) = \sum_{j=0}^{n} c_j \phi_j(x)$$

$$\phi_0(x) = 1; \phi_1(x) = x - x_0; \phi_2(x) = (x - x_0)(x - x_1)$$
$$\phi_3(x) = (x - x_0)(x - x_1)(x - x_2)$$

- Applying the equality of function value and the polynomial value at $x=x_0$: $c_0=f(x_0)$.
- At $x=x_1$:

$$f(x_1) = c_0 + c_1(x_1 - x_0) \Rightarrow c_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

Newton's divided difference

- At x=x_2:

$$f(x_2) = c_0 + c_1(x_2 - x_0) + c_2(x_2 - x_0)(x_2 - x_1)$$

$$\Rightarrow c_2 = \frac{f(x_2) - f(x_0) - (x_2 - x_0)\dfrac{f(x_1) - f(x_0)}{x_1 - x_0}}{(x_2 - x_0)(x_2 - x_1)}$$

$$= \frac{\dfrac{f(x_2) - f(x_1)}{x_2 - x_1} - \dfrac{f(x_1) - f(x_0)}{x_1 - x_0}}{(x_2 - x_0)}$$

Newton's divided difference

- The divided difference notation:

$$f\left[x_j, x_i\right] = \frac{f(x_j) - f(x_i)}{x_j - x_i} = f\left[x_i, x_j\right]$$

$$f\left[x_k, x_j, x_i\right] = \frac{f\left[x_k, x_j\right] - f\left[x_j, x_i\right]}{x_k - x_i} = f\left[x_i, x_j, x_k\right] = \ldots$$

$$f\left[x_n, x_{n-1}, \ldots, x_2, x_1, x_0\right] = \frac{f\left[x_n, x_{n-1}, \ldots, x_2, x_1\right] - f\left[x_{n-1}, \ldots, x_2, x_1, x_0\right]}{x_n - x_0}$$

- First divided difference, second,…,nth

Newton's divided difference

- The ith coefficient is then given by the ith divided difference:

$$c_0 = f(x_0); c_1 = f\left[x_1, x_0\right]; c_2 = f\left[x_2, x_1, x_0\right]; \ldots$$

$$c_n = f\left[x_n, x_{n-1}, \ldots, x_2, x_1, x_0\right]$$

- If hand-computed: easier in a tabular form

x	f(x)	f[x1,x0]	f[x2,x1,x0]
0	1		
		2	
1	3		1
		4	
2	7		

$$c_0 = 1; c_1 = 2; c_2 = 1$$

$$f_2(x) = 1 + 2(x-0) + 1(x-0)(x-1)$$
$$= 1 + x + x^2$$

Newton's divided difference: Error

- The remainder may be written as:

$$R_n(x) = f(x) - f_n(x) = \phi_{n+1}(x) f\left[x, x_n, x_{n-1}, \ldots, x_2, x_1, x_0\right]$$

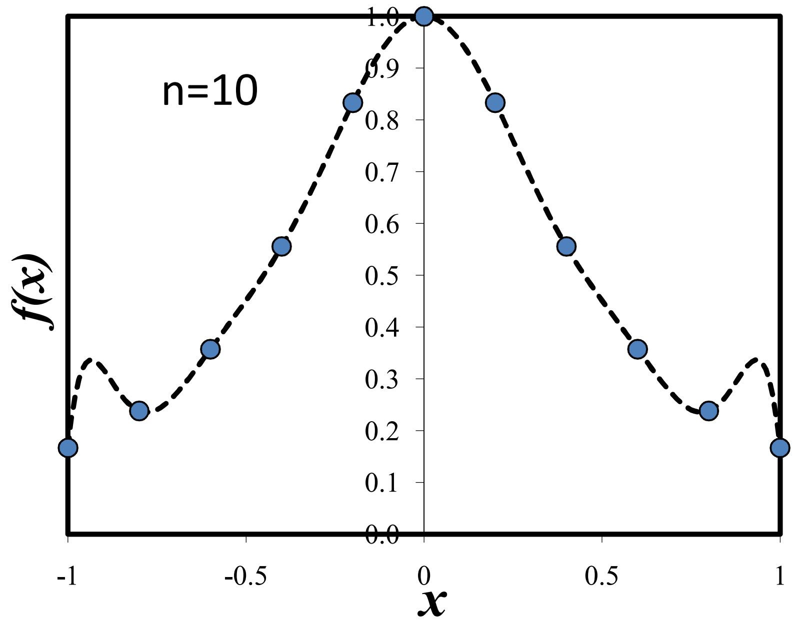
where

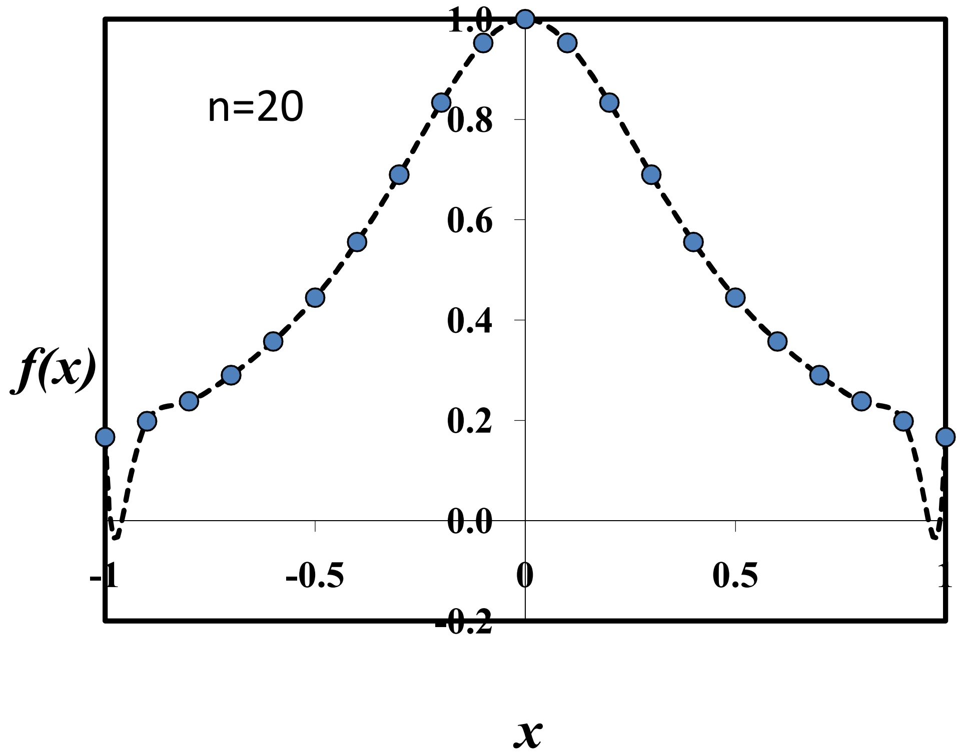
$$\phi_{n+1}(x) = (x - x_0)(x - x_1)(x - x_2)\ldots(x - x_n)$$

- Since f(x) is not known, we may approximate it by using another point (x_{n+1},f(x_{n+1})) as

$$R_n(x) \cong f_{n+1}(x) - f_n(x) = \phi_{n+1}(x) f\left[x_{n+1}, x_n, x_{n-1}, \ldots, x_2, x_1, x_0\right]$$

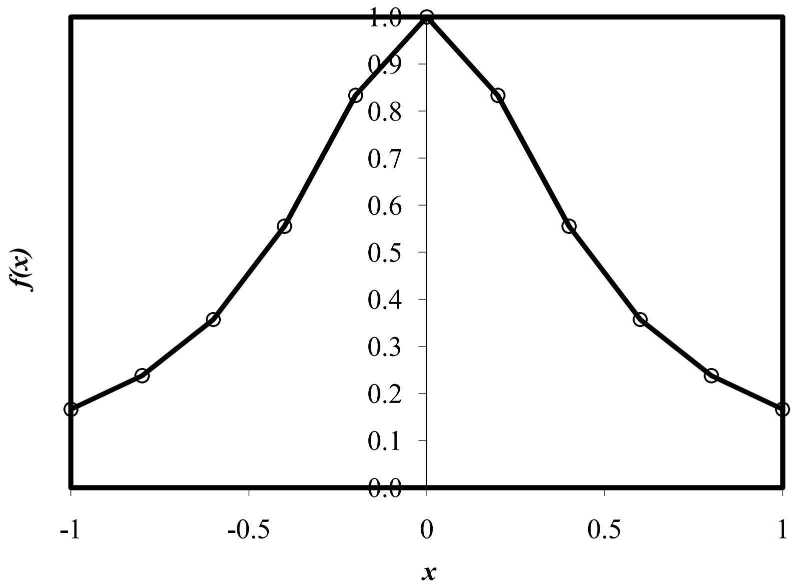
Interpolation: Runge phenomenon





Spline Interpolation

- Using ***piece-wise*** polynomial interpolation
- Given $\left(x_k, f\left(x_k\right)\right) \quad k = 0,1,2,\ldots,n$
- Interpolate using "different" polynomials between smaller segments
- Easiest: Linear between each successive pair
- Problem: First and higher derivatives would be discontinuous



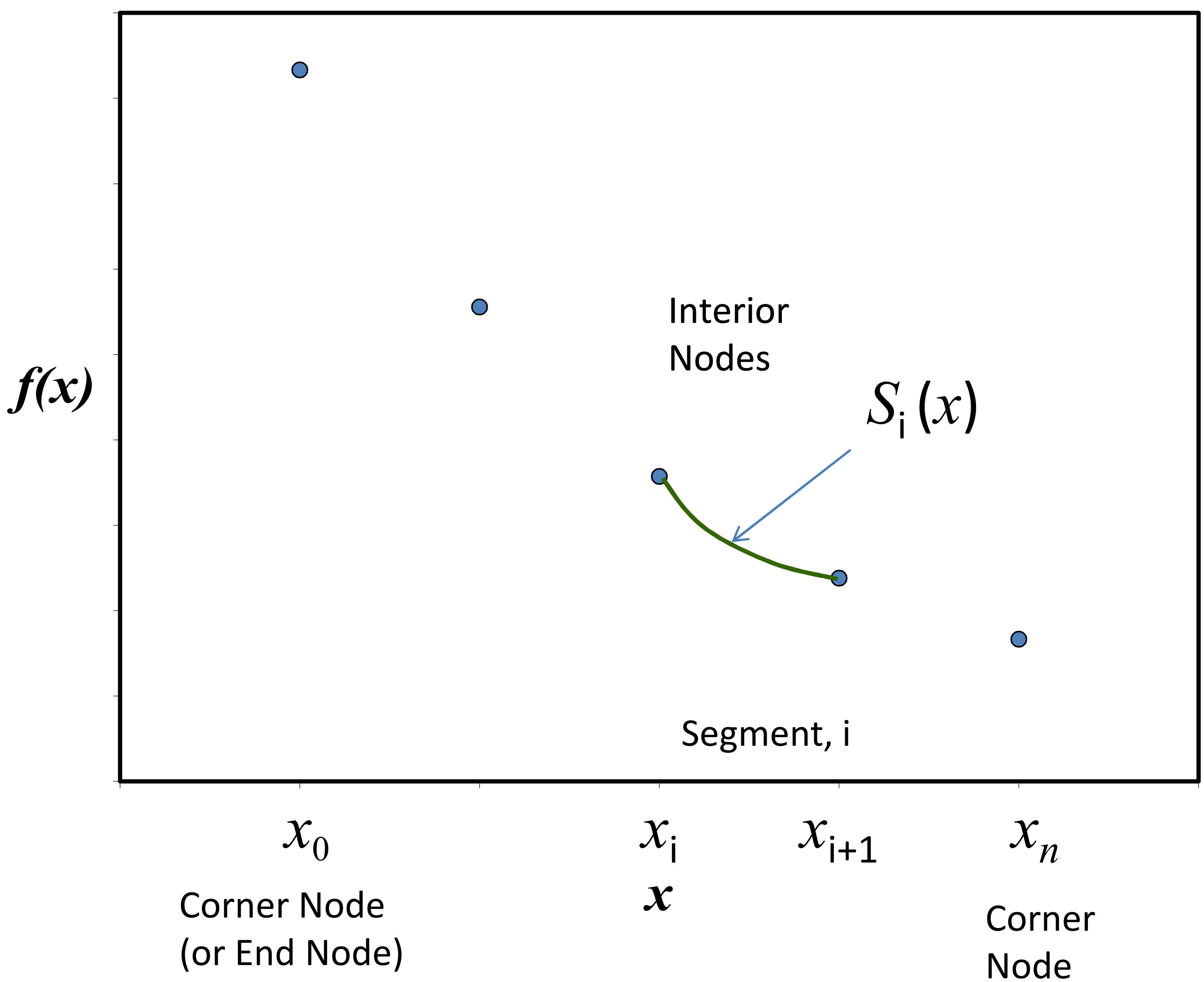
Spline Interpolation

- Most common: Cubic spline
- Given $\left(x_k, f\left(x_k\right)\right) \quad k = 0,1,2,\ldots,n$
- Interpolate using the cubic splines:

$$S_i(x) = c_{0,i} + c_{1,i}\left(x - x_i\right) + c_{2,i}\left(x - x_i\right)^2 + c_{3,i}\left(x - x_i\right)^3$$

between the points x_i and x_{i+1} (i=0,1,2,…,n-1)

- 4 “unknown” coefficients. Two obvious conditions are: $S_i(x_i)=f(x_i)$ and $S_i(x_{i+1})=f(x_{i+1})$
- 2 “degrees of freedom” in each “segment”



f(x)
Interior
Nodes
S_i (x)
Segment, i
x_0
x_i
x_i+1
x_n
x
Corner Node
(or End Node)
Corner
Node

Spline Interpolation

- Total *n* segments => 2*n* d.o.f
- Equality of first and second derivative at *interior nodes* : 2(*n*-1) constraints
- Need 2 more constraints (discussed later)!
- How to obtain the coefficients?
- The second derivative of the cubic spline is linear within a segment. Write it as

$$S_i''(x) = \frac{1}{x_{i+1} - x_i}\left[\left(x_{i+1} - x\right)S_i''(x_i) + \left(x - x_i\right)S_i''(x_{i+1})\right]$$

Spline Interpolation

- Integrate it twice:

$$S_i(x) = \frac{1}{6(x_{i+1}-x_i)}\left[(x_{i+1}-x)^3 S_i''(x_i) + (x-x_i)^3 S_i''(x_{i+1})\right] + C_1 x + C_2$$

and equating the function values at nodes:

$$f(x_i) = \frac{(x_{i+1}-x_i)^2 S_i''(x_i)}{6} + C_1 x_i + C_2$$

$$f(x_{i+1}) = \frac{(x_{i+1}-x_i)^2 S_i''(x_{i+1})}{6} + C_1 x_{i+1} + C_2$$

Spline Interpolation

- Resulting in

$$S_i(x) = \frac{(x_{i+1} - x)^3 S_i''(x_i) + (x - x_i)^3 S_i''(x_{i+1})}{6(x_{i+1} - x_i)}$$
$$+ \left[\frac{f(x_i)}{x_{i+1} - x_i} - \frac{(x_{i+1} - x_i) S_i''(x_i)}{6} \right] (x_{i+1} - x)$$
$$+ \left[\frac{f(x_{i+1})}{x_{i+1} - x_i} - \frac{(x_{i+1} - x_i) S_i''(x_{i+1})}{6} \right] (x - x_i)$$

- How to find the nodal values of *S''* ?

Spline Interpolation

- Continuity of first derivatives: $S_i'(x_i) = S_{i-1}'(x_i)$

$$S_i'(x_i) = -\frac{\left(x_{i+1} - x_i\right)S_i''(x_i)}{3} - \frac{\left(x_{i+1} - x_i\right)S_i''(x_{i+1})}{6} + \frac{f\left(x_{i+1}\right) - f\left(x_i\right)}{x_{i+1} - x_i}$$

$$S_{i-1}'(x_i) = \frac{\left(x_i - x_{i-1}\right)S_{i-1}''(x_i)}{3} + \frac{\left(x_i - x_{i-1}\right)S_{i-1}''(x_{i-1})}{6} + \frac{f\left(x_i\right) - f\left(x_{i-1}\right)}{x_i - x_{i-1}}$$

- Second derivative is also continuous
- We get a *tridiagonal* system

$$\begin{aligned}\left(x_i - x_{i-1}\right)S_{i-1}'' + 2\left(x_{i+1} - x_{i-1}\right)S_i'' + \left(x_{i+1} - x_i\right)S_{i+1}''\\ = 6\frac{f\left(x_{i+1}\right) - f\left(x_i\right)}{x_{i+1} - x_i} - 6\frac{f\left(x_i\right) - f\left(x_{i-1}\right)}{x_i - x_{i-1}}\end{aligned}$$

Spline Interpolation

- What are the 2 more required constraints?
  - Clamped: The function is clamped on each corner node forcing both ends to have some **known** fixed slope, say, s_0 and s_n. This implies $S_0' = s_0$ and $S_n' = s_n$
  - Natural: Curvature at the corner nodes is zero, i.e.,
$$S_0'' = S_n'' = 0$$
  - Not-a-knot: The first and last **interior** nodes have C^3 continuity, i.e., these do not act as a knot, i.e.,
$$S_0(x) \equiv S_1(x) \quad \text{and} \quad S_{n-2}(x) \equiv S_{n-1}(x)$$
  - For periodic functions, $S_0' = S_n'$ and $S_0'' = S_n''$

Spline Interpolation: Example

- From the following data, estimate *f*(2.6)

x	0	1	2	3	4
f(x)	1	0.5	0.2	0.1	0.05882

- The tridiagonal equations are (using natural spline, $S_0'' = S_4'' = 0$):

$$\begin{bmatrix} 4 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 4 \end{bmatrix} \begin{Bmatrix} S_1'' \\ S_2'' \\ S_3'' \end{Bmatrix} = \begin{Bmatrix} 1.2 \\ 1.2 \\ 0.35294 \end{Bmatrix}$$

- Solution : $S_1'' = 0.2420; S_2'' = 0.2319; S_3'' = 0.03025$

Spline Interpolation: Example

- The desired spline (between 2 and 3) is

$$S_2(x)=\frac{(x_3-x)^3 S_2''+(x-x_2)^3 S_3''}{6(x_{i+1}-x_i)}$$

$$+\left[\frac{f(x_2)}{x_3-x_2}-\frac{(x_3-x_2)S_2''}{6}\right](x_3-x)$$

$$+\left[\frac{f(x_3)}{x_3-x_2}-\frac{(x_3-x_2)S_3''}{6}\right](x-x_2)$$

- Putting values

$$S_2(x)=\frac{(3-x)^3 0.2319+(x-2)^3 0.03025}{6}+\left[0.2-\frac{0.2319}{6}\right](3-x)+\left[0.1-\frac{0.03025}{6}\right](x-2)$$

- At x=2.6, $f(x)$= 0.1251