Interpolation

- In numerical methods, interpolation is a process for finding a new value between known data points.
- It involves constructing a simple function that passes through the existing data points to estimate the value of the dependent variable for an intermediate value of the independent variable.
- This is useful for estimating missing data or for simplifying complex functions with a simpler one, such as a polynomial.

Common Type:

- Polynomial Interpolation (Lagrange’s Interpolation, Newton’s Interpolation)
- Piecewise Interpolation (Linear, Quadratic, Cubic)
- Spline Interpolation (Linear, Quadratic, Cubic)

Interpolation with Equal Intervals:

Finite difference:

- It measures how the function f(x) changes between those known points.
- Foundation for interpolation formulas.
- Interpolation uses those changes (Finite difference) to estimate new values between known data points.

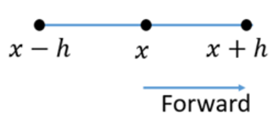
❖ **Forward difference:**

Used in Newton’s Forward Interpolation formula.

✓Forward difference is denoted by (delta) Δ

✓Formula of forward difference is

$$\Delta f(x) = f(x+h) - f(x)$$



If $y_0, y_1, y_2, \ldots, y_n$ denote a set of values of any function y = f(x), then y_1-y_0, $y_2- y_1$, y_3-y_2, …, y_n-y_{n-1} are called the differences of the function y.

$$\Delta y_0 = y_1 - y_0$$

$$\Delta y_1 = y_2 - y_1$$

$$\Delta y_n = y_{n+1} - y_n$$

$$\Delta f(x) = f(x+h) - f(x)$$

Thus,

$$\Delta^2 y_0 = \Delta y_1 - \Delta y_0 = y_2 - y_1 - (y_1 - y_0) = y_2 - 2y_1 + y_0$$

$$\Delta^3 y_0 = \Delta^2 y_1 - \Delta^2 y_0 = y_3 - 2y_2 + y_1 - (y_2 - 2y_1 + y_0)$$
$$= y_3 - 3y_2 + 3y_1 - y_0,$$

and

$$\Delta^4 y_1 = \Delta^3 y_1 - \Delta^3 y_0 = y_4 - 3y_3 + 3y_2 - y_1 - (y_3 - 3y_2 + 3y_1 - y_0)$$
$$= y_4 - 4y_3 + 6y_2 - 4y_1 + y_0$$

Forward (Diagonal) Difference Table:

x	y	Δ	Δ^2	Δ^3	Δ^4	Δ^5	Δ^6
x_0	y_0						
		Δy_0					
x_1	y_1		$\Delta^2 y_0$				
		Δy_1		$\Delta^3 y_0$			
x_2	y_2		$\Delta^2 y_1$		$\Delta^4 y_0$		
		Δy_2		$\Delta^3 y_1$		$\Delta^5 y_0$	
x_3	y_3		$\Delta^2 y_2$		$\Delta^4 y_1$		$\Delta^6 y_0$
		Δy_3		$\Delta^3 y_2$		$\Delta^5 y_1$	
x_4	y_4		$\Delta^2 y_3$		$\Delta^4 y_2$		
		Δy_4		$\Delta^3 y_3$			
x_5	y_5		$\Delta^2 y_4$				
		Δy_5					
x_6	y_6						

Example:

Given the set of values of (x, y) are (10, 19.97), (15, 21.51), (20, 22.47), (25, 23.52), (30, 24.65), (35, 25.89). Construct the forward difference table and find the values at $\Delta\, y_{10}$, $\Delta^2 y_{20}$, $\Delta^3 y_{15}$, $\Delta^5 y_{10}$.

x	10	15	20	25	30	35
y	19.97	21.51	22.47	23.52	24.65	25.89

Solution: Forward difference table:

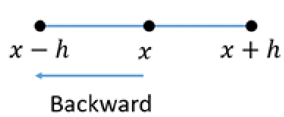
x	y	Δ	Δ^2	Δ^3	Δ^4	Δ^5
10	19.97					
		1.54				
15	21.51		-0.58			
		0.96		0.67		
20	22.47		0.09		-0.68	
		1.05		-0.01		0.72
25	23.52		0.08		0.04	
		1.13		0.03		
30	24.65		0.11			
		1.24				
35	25.89					

❖ **Backward (Horizontal) Differences:**

✓ Backward difference is denoted by (nabla) ∇

✓ Formula of Backward difference is

$$\nabla f(x) = f(x) - f(x-h)$$



The y_1-y_0, y_2-y_1, y_3-y_2, …, y_n-y_{n-1} are called the backward or horizontal differences of the function y, if they denoted by ∇y_1, ∇y_2, …, ∇y_n, .

$$\nabla y_1 = y_1 - y_0$$

$$\nabla y_2 = y_2 - y_1$$

$$\nabla y_n = y_n - y_{n-1}$$

Backward (Horizontal) Difference Table:

x	y	∇	∇^2	∇^3	∇^4	∇^5	∇^6
x_0	y_0						
x_1	y_1	∇y_1					
x_2	y_2	∇y_2	$\nabla^2 y_2$				
x_3	y_3	∇y_3	$\nabla^2 y_3$	$\nabla^3 y_3$			
x_4	y_4	∇y_4	$\nabla^2 y_4$	$\nabla^3 y_4$	$\nabla^4 y_4$		
x_5	y_5	∇y_5	$\nabla^2 y_5$	$\nabla^3 y_5$	$\nabla^4 y_5$	$\nabla^5 y_5$	
x_6	y_6	∇y_6	$\nabla^2 y_6$	$\nabla^3 y_6$	$\nabla^4 y_6$	$\nabla^5 y_6$	$\nabla^6 y_6$

Example:

Given the set of values

x	10	15	20	25	30	35
y	19.97	21.51	22.47	23.52	24.65	25.89

form the difference table and write down the values of

$$\nabla y_{20}, \quad \nabla^2 y_{25}, \quad \nabla^3 y_{30} \quad and \quad \nabla^5 y_{35}$$

Solution:

x	y	∇y	∇²y	∇³y	∇⁴y	∇⁵y
10	19.97	—	—	—	—	—
15	21.51	1.54	—	—	—	—
20	22.47	0.96	−0.58	—	—	—
25	23.52	1.05	0.09	0.67	—	—
30	24.65	1.13	0.08	−0.01	−0.68	—
35	25.89	1.24	0.11	0.03	0.04	0.72

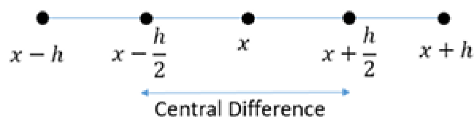
❖ Central Differences:

✓ Central Operator is denoted by (small delta) [δ]

✓ Formula of Central Operator is

$$\delta f(x) = f\left(x + \frac{h}{2}\right) - f\left(x - \frac{h}{2}\right)$$

$$\delta = E^{\frac{1}{2}} - E^{-\frac{1}{2}} \quad (\because E^n f(x) = f(x + nh))$$



Central Difference Table:

The central difference operator δ is defined by the relations

$$y_1 - y_0 = \delta y_{1/2}, \quad y_2 - y_1 = \delta y_{3/2}, \ldots\ldots, y_n - y_{n-1} = \delta y_{n-1/2}$$

Central difference table

x	y	δy	$\delta^2 y$	$\delta^3 y$	$\delta^4 y$	$\delta^5 y$
x_0	y_0					
		$\delta y_{1/2}$				
x_1	y_1		$\delta^2 y_1$			
		$\delta y_{3/2}$		$\delta^3 y_{3/2}$		
x_2	y_2		$\delta^2 y_2$		$\delta^4 y_2$	
		$\delta y_{5/2}$		$\delta^3 y_{5/2}$		$\delta^5 y_{5/2}$
x_3	y_3		$\delta^2 y_3$		$\delta^4 y_3$	
		$\delta y_{7/2}$		$\delta^3 y_{7/2}$		
x_4	y_4		$\delta^2 y_4$			
		$\delta y_{9/2}$				
x_5	y_5					

Example: (Home-work)

Given the set of values

x	10	15	20	25	30	35
y	19.97	21.51	22.47	23.52	24.65	25.89

The General Form of Newton's Interpolating Polynomial

Type	Formula Type	Used For
Newton's Forward / Backward Interpolation	Uses finite differences (Δ or ∇)	Equal intervals (i.e., $h = x_{i+1} - x_i$ is constant)
Newton's Divided Difference Interpolation	Uses divided differences	Unequal intervals (i.e., h is not constant)

Divided Difference:

If we want to compute an n-th order polynomial of the form

$$f_n(x) = b_0 + b_1(x - x_0) + \cdots + b_n(x - x_0)(x - x_1) \ldots (x - x_{n-1})$$

In this case we need to compute $n + 1$ coefficients $b_0, b_1, \ldots b_n$ which can be found as follows:

$$\begin{aligned}
b_0 &= f(x_0) \\
b_1 &= f[x_1, x_0] \\
b_2 &= f[x_2, x_1, x_0] \\
&. \\
&. \\
b_n &= f[x_n, x_{n-1}, \ldots x_1, x_0]
\end{aligned}$$

The bracketed function evaluations are finite divide differences. For example

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

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

$$f[x_n, x_{n-1}. \ldots x_1, x_0] = \frac{f[x_n, x_{n-1}, \ldots x_1] - f[x_{n-1}, x_{n-2}, \ldots, x_0]}{x_n - x_0}$$

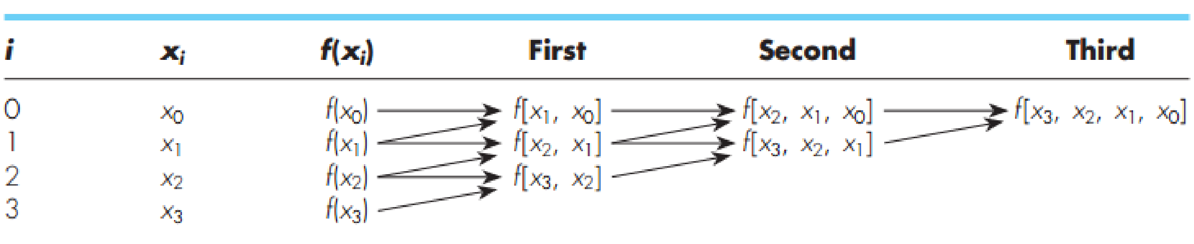


FIGURE 18.5
Graphical depiction of the recursive nature of finite divided differences.

Now we can compute our interpolating polynomial as

$$f_n(x) = b_0 + f[x_1, x_0](x - x_0) + (x - x_0)(x - x_1)f[x_2, x_1, x_0] + \cdots$$
$$+ (x - x_0)(x - x_1) \ldots (x - x_{n-1})f[x_n, x_{n-1}, \ldots, x_0] \quad (14)$$

Interpolation with Unequal Intervals:

Newton's Divided Difference:

- Newton's divided difference is a flexible interpolation formula.
- It builds the interpolation polynomial step by step using divided differences.
- It's useful when x-values are not equally spaced.

Example:

Find Newton's interpolating polynomial using Newton's divided difference approach to approximate a function whose 5 data points are given below.

x	$f(x)$
2.0	0.85467
2.3	0.75682
2.6	0.43126
2.9	0.22364
3.2	0.08567

i	x_i	$f[x_i]$	$f[x_{i-1}, x_i]$	$f[x_{i-2}, x_{i-1}, x_i]$	$f[x_{i-3}, \cdots, x_i]$	$f[x_{i-4}, \cdots, x_i]$
0	**2.0**	**0.85467**				
			-0.32617			
1	**2.3**	**0.75682**		**-1.26505**		
			-1.08520		**2.13363**	
2	**2.6**	**0.43126**		**0.65522**		**-2.02642**
			-0.69207		**-0.29808**	
3	**2.9**	**0.22364**		**0.38695**		
			-0.45990			
4	**3.2**	**0.08567**				

The 5 coefficients of the Newton's interpolating polynomial are:

$$a_0 = f[x_0] = 0.85467$$

$$a_1 = f[x_0, x_1] = -0.32617$$

$$a_2 = f[x_0, x_1, x_2] = -1.26505$$

$$a_3 = f[x_0, x_1, x_2, x_3] = 2.13363$$

$$a_4 = f[x_0, x_1, x_2, x_3, x_4] = -2.02642$$

$$\begin{aligned} P(x) = a_0 &+ a_1(x-x_0) \\ &+ a_2(x-x_0)(x-x_1) \\ &+ a_3(x-x_0)(x-x_1)(x-x_2) \\ &+ a_4(x-x_0)(x-x_1)(x-x_2)(x-x_3) \end{aligned}$$

$$\begin{aligned} P(x) = 0.85467 &- 0.32617(x-2.0) \\ &-1.26505(x-2.0)(x-2.3) \\ &+ 2.13363(x-2.0)(x-2.3)(x-2.6) \\ &-2.02642(x-2.0)(x-2.3)(x-2.6)(x-2.9) \end{aligned}$$

Now, P(x) can be used to estimate the value of the function f(x) say at x = 2.8.

$$\begin{aligned} P(2.8) = 0.85467 &- 0.32617(2.8-2.0) \\ &-1.26505(2.8-2.0)(2.8-2.3) \\ &+ 2.13363(2.8-2.0)(2.8-2.3)(2.8-2.6) \\ &-2.02642(2.8-2.0)(2.8-2.3)(2.8-2.6)(2.8-2.9) \end{aligned}$$

$$f(2.8) \approx P(2.8) = 0.275$$

Lagrange Interpolating Polynomial

The Lagrange interpolating polynomial is simply a reformulation of the Newton polynomial that avoids the computation of divided differences. It can be represented as

$$f_n(x) = \sum_{i=0}^{n} L_i(x) f(x_i) \quad (19)$$

Where

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} \quad (20)$$

In summary,

a) for cases where the order of the polynomial is unknown, the Newton method has advantages because of the insight it provides into the behavior.
b) In addition, the error estimate represented by can usually be integrated easily into the Newton computation.
c) When only one interpolation is to be performed, the Lagrange and Newton formulations require comparable computational effort. However, the Lagrange version is somewhat easier to program. Because it does not require computation and storage of divided differences, the Lagrange form is often used when the order of the polynomial is known a priori.

Spline Interpolation

In the previous sections, $n - 1$th-order polynomials were used to interpolate between n data points. However, there are cases where these functions can lead to erroneous results because of round-off error and overshoot. An alternative approach is to apply lower-order polynomials to subsets of data points. Such connecting polynomials are called *spline functions*. For example, third-order curves employed to connect each pair of data points are called cubic splines.

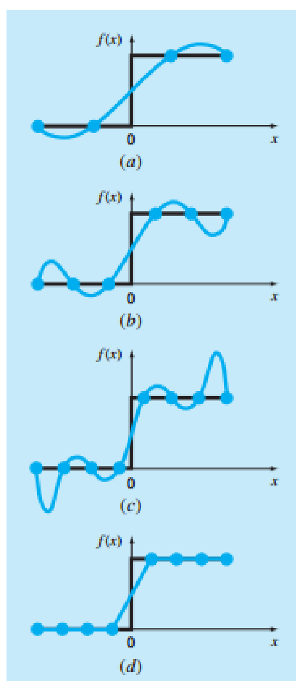


FIGURE 18.14
A visual representation of a situation where the splines are superior to higher-order interpolating polynomials. The function to be fit undergoes an abrupt increase at $x = 0$. Parts (a) through (c) indicate that the abrupt change induces oscillations in interpolating polynomials. In contrast, because it is limited to third-order curves with smooth transitions, a linear spline (d) provides a much more acceptable approximation.

Linear Splines

The simplest connection between two points is a straight line. The first-order splines for a group of ordered data points can be defined as a set of linear functions

$$
\begin{aligned}
&f(x) = f(x_0) + m_0(x - x_0) \; x_0 \le x \le x_1 \\
&f(x) = f(x_1) + m_1(x - x_1) \; x_1 \le x \le x_2 \\
&. \\
&. \\
&f(x) = f(x_{n-1}) + m_{n-1}(x - x_{n-1}) \; x_{n-1} \le x \le x_n
\end{aligned}
$$

where m_i is the slope of the straight line connecting the points:

$$m_i = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}$$

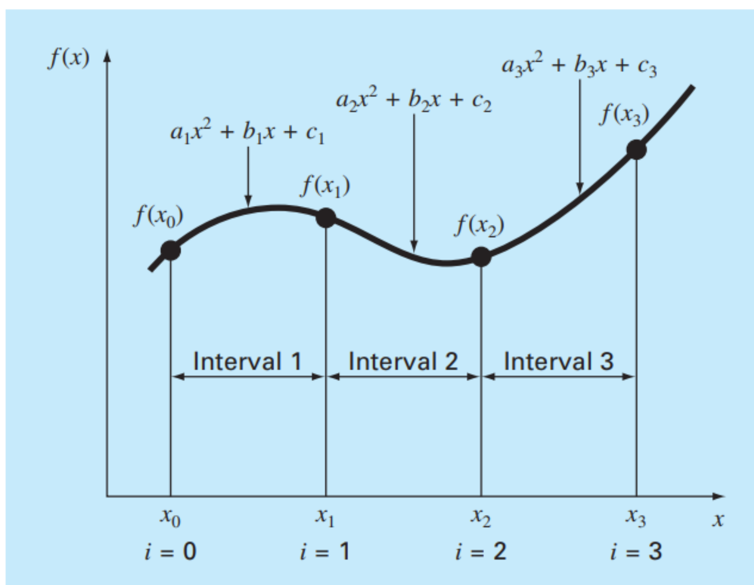
Quadratic Splines

The objective in quadratic splines is to derive a second-order polynomial for each interval between data points. The polynomial for each interval can be represented generally as

$$s_i(x) = a_i x^2 + b_i x + c_i$$

Figure 18.17 has been included to help clarify the notation. For $(n+1)$ data points ($i = 0, 1, 2, \dots n$), there are n intervals and, consequently, $3n$ unknown constants (the a's, b's, and c's) to evaluate. Therefore, $3n$ equations or conditions are required to evaluate the unknowns. These are:

FIGURE 18.17
Notation used to derive quadratic splines. Notice that there are n intervals and $n + 1$ data points. The example shown is for $n = 3$.



1. The function values of adjacent polynomials must be equal at the interior knots. This condition can be represented as

$$
\begin{aligned}
a_{i-1}x_{i-1}^2 + b_{i-1}x_{i-1} + c_{i-1} = f(x_{i-1}) \\
a_i x_{i-1}^2 + b_i x_{i-1} + c_i = f(x_{i-1})
\end{aligned}
$$

For $i = 2$ to n. Because only interior knots are used, the equations above provide $(n-1)$ conditions for a total of $2n - 2$ conditions.

2. The first and last functions must pass through the end points. This adds two additional equations:

$$a_1 x_0^2 + b_1 x_0 + c_1 = f(x_0)$$
$$a_n x_n^2 + b_n x_n + c_n = f(x_n)$$

For a total of $2n - 2 + 2 = 2n$ conditions.

3. The first derivatives at the interior knots must be equal. The first derivative of the quadratic spline is

$$f'(x) = 2ax + b$$

Therefore, the conditions can be represented as

$$2a_{i-1}x_{i-1} + b_{i-1} = 2a_i x_{i-1} + b_i$$

For $i = 2$ to n. This provides another $n - 1$ conditions for a total $2n + n - 1 = 3n - 1$ so we are one condition short. Unless we have some additional information regarding the functions or their derivatives, we must make an arbitrary choice to successfully compute the constants. Although there are a number of different choices that can be made, we select the following:

4. Assume that the second derivative is zero at the first point. Because the second derivative of the spline is $2a_i$, this condition can be expressed mathematically as

$$a_1 = 0$$

The visual interpretation of this condition is that the first two points will be connected by a straight line.

Example:

Formulate the system of equations required to fit quadratic spline functions to the given dataset. Can you express the problem in the matrix form $A\vec{x}=\vec{b}$?

x	1.0	2.0	4.0	5.0
$f(x)$	2.0	3.0	1.5	2

Step 1: Define the splines

We have 4 points → 3 intervals:

- $S_1(x) = a_1x^2 + b_1x + c_1$ on $[1.0, 2.0]$
- $S_2(x) = a_2x^2 + b_2x + c_2$ on $[2.0, 4.0]$
- $S_3(x) = a_3x^2 + b_3x + c_3$ on $[4.0, 5.0]$

Unknowns: 9 coefficients in total.

Step 2: Conditions to construct the system

- Function value matching (6 equations)
- First derivative continuity (2 equations)
- One extra condition (e.g. natural spline: second derivative at first point = 0)

(a) Interpolation (6 equations)

1. $S_1(1.0) = 2.0$

$$a_1(1)^2 + b_1(1) + c_1 = 2.0 \Rightarrow a_1 + b_1 + c_1 = 2.0$$

2. $S_1(2.0) = 3.0$

$$4a_1 + 2b_1 + c_1 = 3.0$$

3. $S_2(2.0) = 3.0$

$$4a_2 + 2b_2 + c_2 = 3.0$$

4. $S_2(4.0) = 1.0$

$$16a_2 + 4b_2 + c_2 = 1.0$$

5. $S_3(4.0) = 1.0$

$$16a_3 + 4b_3 + c_3 = 1.0$$

6. $S_3(5.0) = 2.0$

$$25a_3 + 5b_3 + c_3 = 2.0$$

(b) First derivative continuity (2 equations)

$$S_i'(x) = 2a_i x + b_i$$

7. At $x = 2.0$:

$$S_1'(2) = S_2'(2) \Rightarrow 4a_1 + b_1 = 4a_2 + b_2 \Rightarrow 4a_1 - 4a_2 + b_1 - b_2 = 0$$

8. At $x = 4.0$:

$$S_2'(4) = S_3'(4) \Rightarrow 8a_2 + b_2 = 8a_3 + b_3 \Rightarrow 8a_2 - 8a_3 + b_2 - b_3 = 0$$

(c) Second derivative constraint (1 equation)

9. Let $S_1''(1.0) = 0 \rightarrow 2a_1 = 0 \Rightarrow a_1 = 0$

Step 3: Write the system $A\vec{x} = \vec{b}$

$$\vec{x} = \begin{bmatrix} a_1 \\ b_1 \\ c_1 \\ a_2 \\ b_2 \\ c_2 \\ a_3 \\ b_3 \\ c_3 \end{bmatrix} \qquad \vec{b} = \begin{bmatrix} 2.0 \\ 3.0 \\ 3.0 \\ 1.0 \\ 1.0 \\ 2.0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Coefficient matrix A:

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 16 & 4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 16 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 25 & 5 & 1 \\ 4 & 1 & 0 & -4 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8 & 1 & 0 & -8 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$