Newton Forward Difference Interpolation

Lecture 23 • Numerical Interpolation

Purpose	Estimate a function between equally spaced tabulated points.
Core formula	Use a polynomial built from the first entries of forward difference columns.
Connection	Build on Lecture 22; backward interpolation follows in Lecture 24.

Learning outcomes

By the end of this lecture, students can:

Derive	Construct the Newton forward polynomial using finite differences.
Compute	Evaluate interpolated values by hand using a difference table.
Assess	Check spacing, estimate interpolation error and implement the method.

When to use the method

Given distinct nodes $x_i = x_0 + ih$ and values $y_i = f(x_i)$.

Equal spacing	$h = x_{i+1} - x_i$ must be constant and nonzero.
Target location	Prefer a target near the beginning of the selected data table.
Unequal spacing	Use Newton divided differences instead (Lecture 25).

Newton forward formula

Define $u = (x - x_0)/h$.

First terms	$P_n(x) = y_0 + u\Delta y_0 + u(u-1)\Delta^2 y_0/2!$
Next term	$+ u(u-1)(u-2)\Delta^3 y_0/3! + \ldots$
General term	Term k = $[u(u-1)\ldots(u-k+1)/k!]\ \Delta^k y_0$.

Derivation — Newton basis

Write a polynomial in falling-factorial form.

Start	$P(u) = c_0 + c_1 u + c_2 u(u-1) + \dots$
At u = 0, 1	$P(0)=y_0 \Rightarrow c_0=y_0$; $P(1)=y_1 \Rightarrow c_1=\Delta y_0$.
At u = 2	$y_2 = y_0 + 2\Delta y_0 + 2c_2 \Rightarrow c_2=\Delta^2 y_0/2!$.

Derivation — general coefficient

The kth forward difference isolates the kth basis coefficient.

Difference identity	$\Delta[u(u-1)\dots(u-k+1)] = k\cdot u(u-1)\dots(u-k+2)$.
Repeat k times	At u=0, higher-degree terms vanish after k differences; lower terms are zero.
Result	$\Delta^k P(0) = k!c_k = \Delta^k y_0$; hence $c_k = \Delta^k y_0/k!$.

Step-by-step algorithm

Choose n+1 consecutive equally spaced samples.

Step 1	Compute h and $u = (x - x_0)/h$; construct forward difference columns.
Step 2	Read the top entry of each column, starting with y_0.
Step 3	Accumulate terms using $w_0 = 1$ and $w_k = w_{k-1}(u - k + 1)/k$.

Example 1 — Quadratic data

Given $f(x)=x^2$ at $x=0,1,2,3,4$.

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
0	0	1	2	0	0
1	1	3	2	0	
2	4	5	2		
3	9	7			
4	16				

Read the first row: $y_0=0$, $\Delta y_0=1$, $\Delta^2 y_0=2$; higher differences vanish.

Example 1 — evaluate at x = 0.5

Using the quadratic table on the previous slide.

Normalize	h=1, x_0=0, u=0.5. Top entries: 0, 1, 2, 0, 0.
Substitute	P(0.5) = 0 + 0.5(1) + 0.5(−0.5)(2)/2.
Answer	P(0.5)=0.25, exactly matching x^2.

Example 1 — Term-by-term evaluation

At u=0.5, multiply each top difference by its basis weight.

k	Top $\Delta^k y_0$	Weight	Term	Sum	
0	0	1	0	0	
1	1	0.5	0.5	0.5	
2	2	−0.125	−0.25	0.25	
3	0	0.0625	0	0.25	
4	0	−0.0390625	0	0.25	

Sum = 0.25. Zero higher differences contribute nothing.

Example 2 — Cubic data

Given $f(x)=x^3$ at $x=0,1,2,3,4$.

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
0	0	1	6	6	0
1	1	7	12	6	
2	8	19	18		
3	27	37			
4	64				

Top row: 0, 1, 6, 6, 0. Use terms through order 3.

Example 2 — evaluate at x = 1.5

Use the cubic table: $y_0=0$, $\Delta y_0=1$, $\Delta^2 y_0=6$, $\Delta^3 y_0=6$.

Normalize	h=1, u=1.5. Fourth difference is zero.
Substitute	$P=1.5 + (1.5\times0.5/2)\times6 + (1.5\times0.5\times(-0.5)/6)\times6$.
Answer	$P=1.5+2.25-0.375=3.375$, exactly matching 1.5^3.

Example 3 — Non-unit step size

Given $f(x)=2x^2+1$ at $x=0,2,4,6$.

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$
0	1	8	16	0
2	9	24	16	
4	33	40		
6	73			

Top row: 1, 8, 16, 0. Always normalize using h=2.

Example 3 — evaluate at x = 1

Non-unit spacing: $h=2$, $x_0=0$; use the previous table.

Normalize	$u=(1-0)/2=0.5$. Top entries are 1, 8, 16, 0.
Substitute	$P=1 + 0.5(8) + [0.5(-0.5)/2]\times16$.
Answer	$P=1+4-2=3$, matching $f(1)=2(1)^2+1$.

Interpolation error and limitations

For f with n+1 continuous derivatives on the node/target interval:

Remainder	$f(x)-P_n(x) = f^{(n+1)}(\xi)\cdot\prod(x-x_i)/(n+1)!$ for some ξ in that interval.
Bound example	For $f=e^x$ on $[0,1]$, linear interpolation at 0.5: $\lvert\text{error}\rvert \le e/8 \approx 0.340$.
Caution	Extrapolation, noisy data and high-degree polynomials may amplify error.

Python implementation

Assume equally spaced x, nonzero h, and matching nonempty y.

Initialize	u=(target-x[0])/h; v=list(y); total=v[0]; w=1.0

Repeat	`for k in range(1,len(y)):`
	`  v=[b-a for a,b in zip(v,v[1:])]; w*= (u-k+1)/k`
	`  total+=w*v[0]`

Output	For x=[0,1,2], y=[0,1,4], target=0.5, total=0.25. O(n²) time, O(n) memory.

Practice questions

Construct the difference table before applying the formula.

Q1	x=[0,1,2,3], y=[1,4,9,16]. Estimate y at x=0.5.
Q2	x=[0,2,4,6], y=[1,9,33,73]. Estimate y at x=3.
Q3	Can standard Newton forward interpolation be used with x=[0,1,3]? Explain.

Solutions and recap

Next lecture: Newton Backward Difference Interpolation.

A1	u=0.5; P=1+0.5(3)+0.5(−0.5)(2)/2=2.25.
A2	u=1.5; P=1+1.5(8)+1.5(0.5)(16)/2=19.
A3	No: step sizes 1 and 2 differ. Use Newton divided differences.