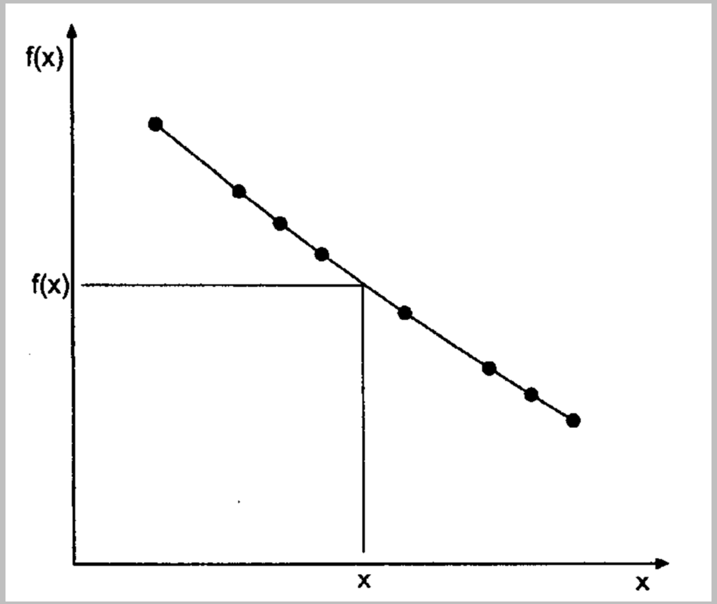
Polynomial Approximation

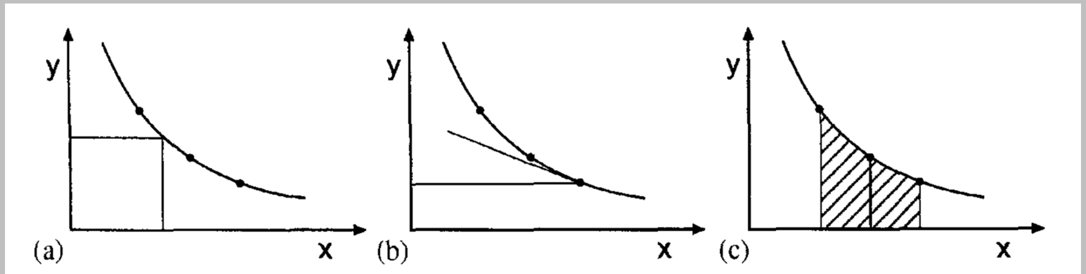
Determination of $f(x)$ at point x not on discrete data points

Interpolation- x inside the interval of data points

Extrapolation - x outside the interval of data points



Differentiation and Integration



Polynomial Fits

Exact Fit

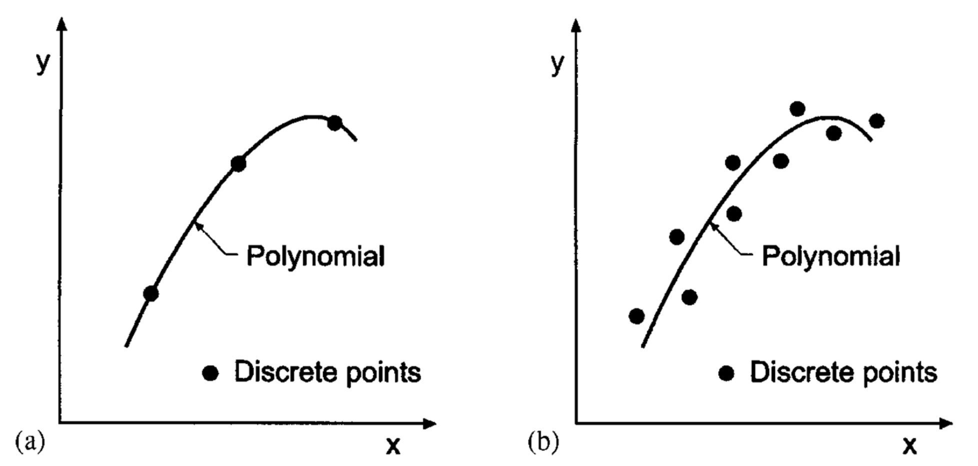
Direct

Lagrange

Approximate Fits

Cubic Spline

Least Square Approximation



Polynomials

Polynomial $P_n(x)$ of nth degree has $(n+1)$ coefficients in the form

$$P_n(x) = a_o + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

For a given x, how many operations to determine $P_n(x)$ from above polynomial?

Nested operations- by successive factorization, $P_n(x)$ written as

$$P_n(x) = a_o + x(a_1 + x\left(a_2 + x(a_3 + \cdots \quad + x(a_{n-1} + a_n x))\right))$$

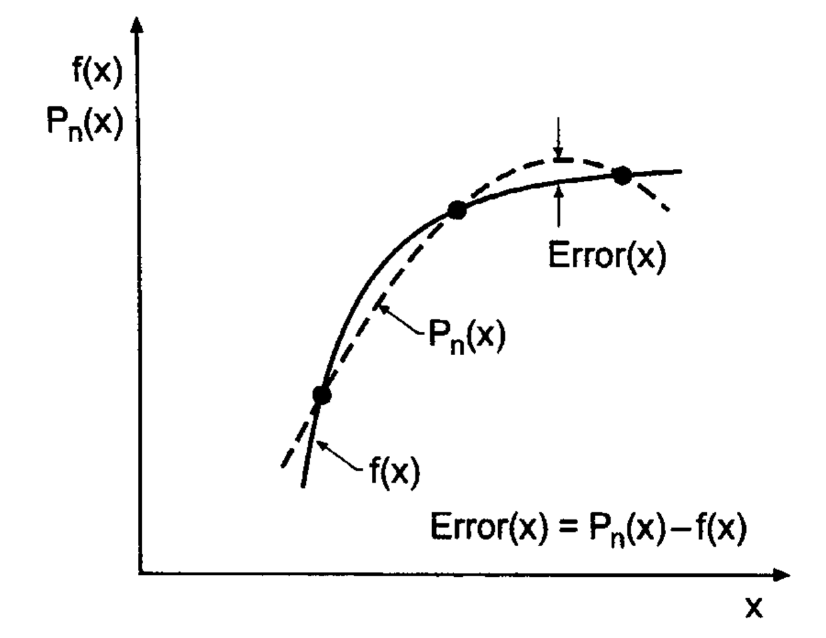
For a given x, how many operations to determine $P_n(x)$ from above nested polynomial?

Weierstrass Theorem

For any small ε, there exist a positive integer N such that a smooth function $f(x)$ in interval $[a, b]$ can be approximated by a polynomial of degree N

$$|f(x) - P_N(x)| < \varepsilon$$

Approximation Error



With $(n+1)$ descrite data points $(x_i, f(x_i))$, $P_n(x)$ can be found that passes through all $(n+1)$ points with $x = x_i$,

$$P_n(x_i) = f(x_i)$$

When $x \neq x_i$, $P_n(x)$ will approximate $f(x)$ with error

$$P_n(x) - f(x) = \frac{1}{n+1}(x - x_o)(x - x_1)(x - x_2) \ldots (x - x_n) f^{n+1}(\xi)$$

$$x_o < \xi < x_n$$

Direct Fit

With $(n+1)$ descrite data points $(x_i, f(x_i))$, an nth order polynomial $P_n(x)$ can be specified that exactly fits all $(n+1)$ points.

To specify $P_n(x) = a_o + a_1 x + a_2 x^2 + \cdots + a_n x^n$, the $(n+1)$ unknown coefficients $(a_o, a_1 \ldots a_n)$ can be determined

$$\begin{cases} a_o + a_1 x_o + a_2 x_o^2 + a_3 x_o^3 + \cdots + a_n x_o^n = f(x_o) \\ a_o + a_1 x_1 + a_2 x_1^2 + a_3 x_1^3 + \cdots + a_n x_1^n = f(x_1) \\ a_o + a_1 x_2 + a_2 x_2^2 + a_3 x_2^3 + \cdots + a_n x_2^n = f(x_2) \\ \cdot \\ \cdot \\ a_o + a_1 x_n + a_2 x_n^2 + a_3 x_n^3 + \cdots + a_n x_n^n = f(x_n) \end{cases}$$

Vector notation

$a = [a_o, a_1, a_2, \ldots a_n]^T$

$$G = \begin{bmatrix} 1 & x_o^1 & x_o^2 & x_o^3 \ldots x_o^n \\ 1 & x_1^1 & x_1^2 & x_1^3 \ldots x_1^n \\ 1 & x_2^1 & x_2^2 & x_2^3 \ldots x_2^n \\ & & \ldots \ldots & \\ 1 & x_n^1 & x_n^2 & x_n^3 \ldots x_n^n \end{bmatrix}$$

$P_n(x) \approx f(x)$ $\quad a = G^{-1}F$ $\quad Ga = F$

$F = [f(x_o), f(x_1), f(x_2), \ldots, f(x_n)]^T$

Lagrange Polynomials

Passing a polynomial through $(n+1)$ points

Two-point fit (x_1, f_1) and (x_2, f_2)

$$f(x) = \frac{x - x_2}{x_1 - x_2} f_1 + \frac{x - x_1}{x_2 - x_1} f_2 = P_2(x)$$

Three-point fit $(x_1, f_1), (x_2, f_2), (x_2, f_2)$

$$f(x) = \frac{x - x_2}{x_1 - x_2} \frac{x - x_3}{x_1 - x_3} f_1 + \frac{x - x_1}{x_2 - x_1} \frac{x - x_3}{x_2 - x_3} f_2 + \frac{x - x_1}{x_3 - x_1} \frac{x - x_2}{x_3 - x_2} f_3 = P_3(x)$$

Determine $P_n(x)$ that passes through 3 points

(0,1); (1, 0.75); (2,0)

Lagrange Polynomials

Passing a polynomial through (n) points

n-point fit

$$P_{n-1}(x) = \frac{(x-x_2)(x-x_3)(x-x_4)\ldots(x-x_n)}{(x_1-x_2)(x_1-x_3)(x_1-x_4)\ldots.(x_1-x_n)} f_1$$

$$\frac{(x-x_1)(x-x_3)(x-x_4)\ldots(x-x_n)}{(x_2-x_1)(x_2-x_3)(x_2-x_4)\ldots.(x_2-x_n)} f_2 +$$

$$\frac{(x-x_1)(x-x_2)(x-x_4)\ldots(x-x_n)}{(x_3-x_1)(x_3-x_2)(x_3-x_4)\ldots.(x_3-x_n)} f_3 +$$

....

$$+\frac{(x-x_1)(x-x_2)(x-x_3)\ldots(x-x_{n-1})}{(x_n-x_1)(x_n-x_2)(x_n-x_3)\ldots.(x_n-x_{n-1})} f_n$$

No of operations

Lagrange

Direct method

Gauss-Seidel

Use Lagrange method for
$P_2(x)$ passing through 3 points

(0,1); (1, 0.75); (2,0)

Navielle Method

This a successive linear interpolation between adjacent points
With n points (x_k, f_k),

$$f_k^o = f_k$$

With

$$f_k^1 = \frac{x-x_k}{x_{k+1}-x_k} f_{k+1}^o + \frac{x-x_{k+1}}{x_k-x_{k+1}} f_k^o$$

And continuing

$$f_k^2 = \frac{x-x_k}{x_{k+1}-x_k} f_{k+1}^1 + \frac{x-x_{k+1}}{x_k-x_{k+1}} f_k^1$$

$$f_k^n = \frac{x-x_k}{x_{k+1}-x_k} f_{k+1}^{n-1} + \frac{x-x_{k+1}}{x_k-x_{k+1}} f_k^{n-1}$$

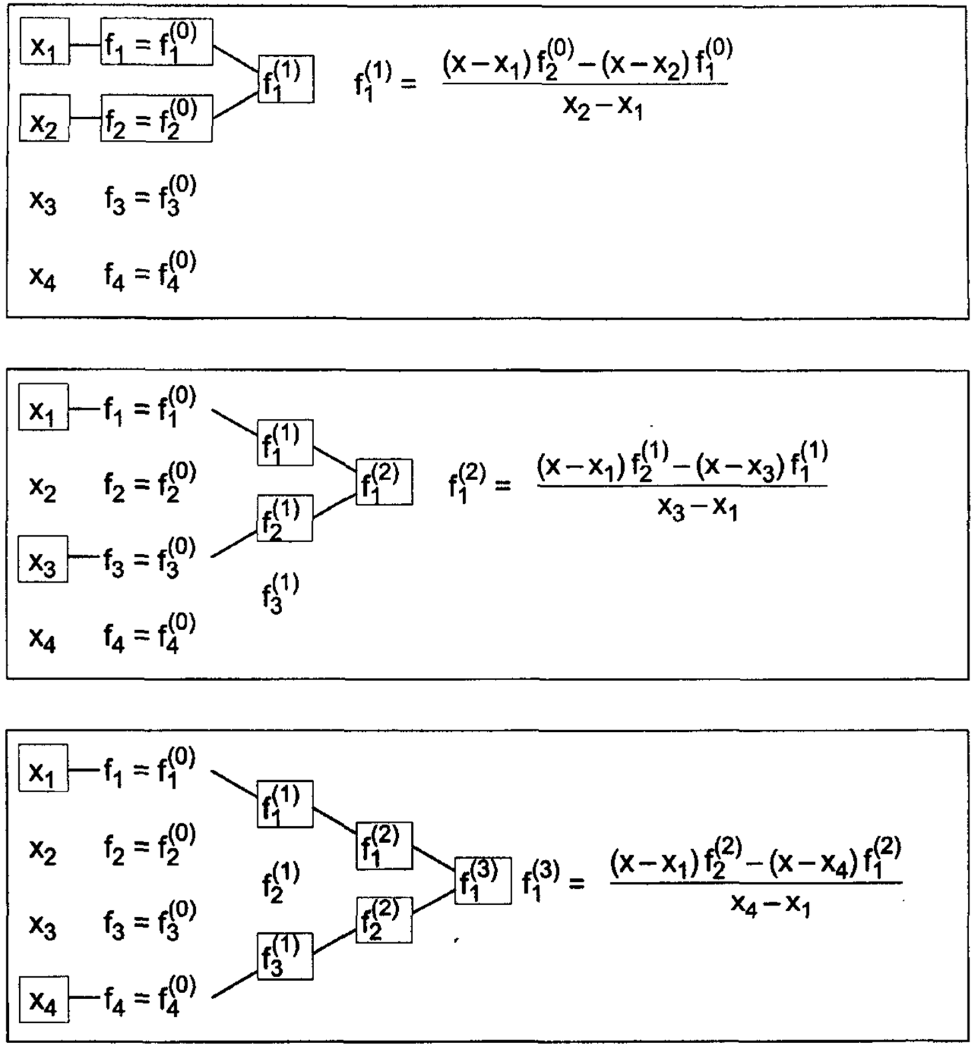
x_i	$f_i^{(0)}$	$f_i^{(1)}$	$f_i^{(2)}$	$f_i^{(3)}$
x_1	$f_1^{(0)}$			
		$f_1^{(1)}$		
x_2	$f_2^{(0)}$		$f_1^{(2)}$	
		$f_2^{(1)}$		$f_1^{(3)}$
x_3	$f_3^{(0)}$		$f_2^{(2)}$	
		$f_3^{(1)}$		
x_4	$f_4^{(0)}$			

$$f_k^n(x) = P_n(x)$$

Advantage:
No. of operations?

Determine $P_n(x)$ that passes through 3 points

(0,1); (1, 0.75); (2,0)



Divided Difference Method

For $(n+1)$ points $(x_i\ ,\ f_i)$, with $f_i\ =\ f(x_i)$, the divided difference is the difference between the f value at two adjacent points divided by the distance between these two points.

Set f_i^o as the original values of f at x_i

$$f_i^o = f_i$$

The next set of divided difference f_i^1 will be

$$f_i^1 = \frac{f_{i+1}^o - f_i^o}{x_{i+1} - x_i}$$

And for kth iteration

$$f_i^{k+1} = \frac{f_{i+1}^k - f_i^k}{x_{i+1} - x_i}$$

At each step of differencing, number of points for next differencing will be reduced by one.

x_i	$f_i^{(0)}$	$f_i^{(1)}$	$f_i^{(2)}$	$f_i^{(3)}$
x_1	$f_1^{(0)}$			
		$f_1^{(1)}$		
x_2	$f_2^{(0)}$		$f_1^{(2)}$	
		$f_2^{(1)}$		$f_1^{(3)}$
x_3	$f_3^{(0)}$		$f_2^{(2)}$	
		$f_3^{(1)}$		
x_4	$f_4^{(0)}$			

Divided Difference Polynomial

Polynomial $P_n(x)$ can be constructed from divided difference values f_i^k

$$\begin{aligned}P_n(x)\\ &= f_i^o + (x - x_o)f_i^1 + (x - x_o)(x - x_1)f_i^2 + \ \ldots\\ &+ (x - x_o)(x - x_1)(x - x_3)..(x - x_n)f_i^n\end{aligned}$$

For any $(n+1)$ points, $P_n(x)$ is unique. The polynomial above passes through all the points, so this expression is equal to Lagrange's.

Differencing

With equal spacing for all points, function f differences correspond to $\frac{df}{dx}$.

Successive differencing of two neighboring values: which neighbor to count for differencing method? Forward, backward or taking a half-step.

Forward differencing Δf_i :

$$\Delta f_i = f_{i+1} - f_i$$

Backward differencing ∇f_i

$$\nabla f_i = f_i - f_{i-1}$$

Centered differencing $\delta f_{i+1/2}$

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

Table of differences

x	f	Δf	$\Delta^2 f$	$\Delta^3 f$
x_0	f_0			
		Δf_0		
x_1	f_1		$\Delta^2 f_0$	
		Δf_1		$\Delta^3 f_0$
x_2	f_2		$\Delta^2 f_1$	
		Δf_2		
x_3	f_3			
		Starting from x_o		

x	f	∇f	$\nabla^2 f$	$\nabla^3 f$
x_{-3}	f_{-3}			
		∇f_{-2}		
x_{-2}	f_{-2}		$\nabla^2 f_{-1}$	
		∇f_{-1}		$\nabla^3 f_0$
x_{-1}	f_{-1}		$\nabla^2 f_0$	
		∇f_0		
x_0	f_0			
		Starting from x_n		

x	f	δf	$\delta^2 f$	$\delta^3 f$
x_{-1}	f_{-1}			
		$\delta f_{-1/2}$		
x_0	f_0		$\delta^2 f_0$	
		$\delta f_{1/2}$		$\delta^3 f_{1/2}$
x_1	f_1		$\delta^2 f_1$	
		$\delta f_{3/2}$		
x_2	f_2			
		Starting from *middle*		

n - number of spacings

$f_o\ f_1\ f_2$ $\quad f_{i-1}\ f_i\ f_{i+1}$ $\quad f_{n-2}\ f_{n-1}\ f_n$

(i-1) i i+1

$$\Delta x = \frac{len}{spacing} = \frac{L}{n} = h$$

x

$x_i = x_o + i * h$ and $f_i = f(x_i)$

$(n+1)$ = number of nodes

Derivatives		first	second	third
x	$f(x)$			
x_0	f_0			
		$(f_1 - f_0)$		
x_1	f_1		$(f_2 - 2f_1 + f_0)$	
		$(f_2 - f_1)$		$(f_3 - 3f_2 + 3f_1 - f_0)$
x_2	f_2		$(f_3 - 2f_2 + f_1)$	
		$(f_3 - f_2)$		
x_3	f_3			

Forward and backward differencing for 3 points

(0,1); (1, 0.75); (2,0)

Round-off errors in differencing

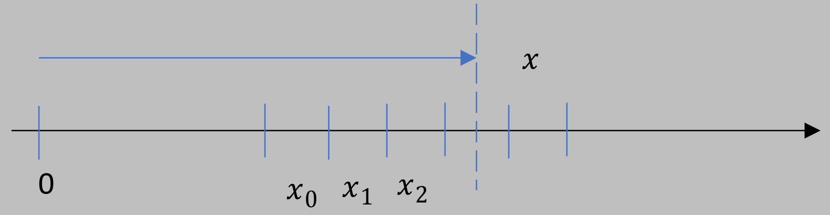
Max. round-off errors in nth derivative = $\pm 2^{n-1}$

x	f	Δf	$\Delta^2 f$	$\Delta^3 f$	$\Delta^4 f$
–	$+1/2$				
		-1			
–	$-1/2$		2		
		1		-4	8
–	$+1/2$		-2		
		-1		4	
–	$-1/2$		2		-8
		1		-4	
–	$+1/2$		-2		
		-1			
–	$-1/2$				

Watch out for round-off errors!

In similar manner constructed for Navielle method, $P_n(x)$ can be constructed from the difference tables and called Newton Polynomials that passes through $(n+1)$ points.

Newton polynomials depends on the difference method, forward, backward and centered.



Newton Polynomials

With $x_i = x_o + ih$, we can define s instead of x

$$s = \frac{x - x_o}{h}$$

$$s_i = i$$

$$x = x_o + sh$$

Newton Forward Differencing Polynomials

$$P_n(x) = f_o + s\Delta f_o + \frac{s(s-1)}{2!}\Delta^2 f_o + \frac{s(s-1)(s-2)}{3!}\Delta^3 f_o + \dots + \frac{ss(s-1)(s-2)\dots(s-n+1)}{n!}\Delta^n f_o$$

Or alternatively in terms of binomial coefficients

$$P_n(x) = f_o + \binom{s}{1}\Delta f_o + \binom{s}{2}\Delta^2 f_o + \binom{s}{3}\Delta^3 f_o + \dots + \binom{s}{n}\Delta^n f_o$$

With binomial coefficients defines as $\binom{s}{i} = \frac{s(s-1)(s-2)\dots(s-[i-1])}{i!}$

Newton Backward Differencing Polynomials

$$P_n(x) = f_o + s\nabla f_o + \frac{s(s+1)}{2!}\nabla^2 f_o + \frac{s(s+1)(s+2)}{3!}\nabla^3 f_o + \ \dots + \frac{ss(s+1)(s+2)\dots(s+n-1)}{n!}\nabla^n f_o$$

Or alternatively in terms of binomial coefficients

$$P_n(x) = f_o + \binom{s^+}{1}\nabla f_o + \binom{s^+}{2}\nabla^2 f_o + \binom{s^+}{3}\nabla^3 f_o + \cdots + \binom{s^+}{n}\nabla^n f_o$$

With binomial coefficients defines as $\binom{s^+}{i} = \frac{s(s+1)(s+2)\dots.(s+[i-1])}{i!}$

Expressions for Newton Centered differencing polynomials use similar formulas.

Determine $P_n(x)$ that passes through 3 points using Newton backward differencing

(0,1); (1, 0.75); (2,0)