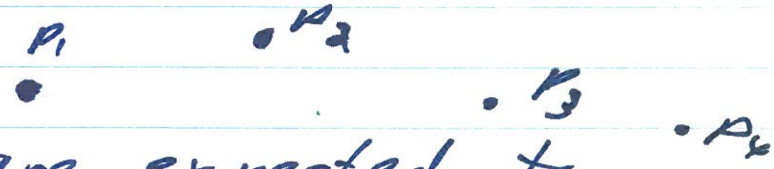
Approximation and Interpolation

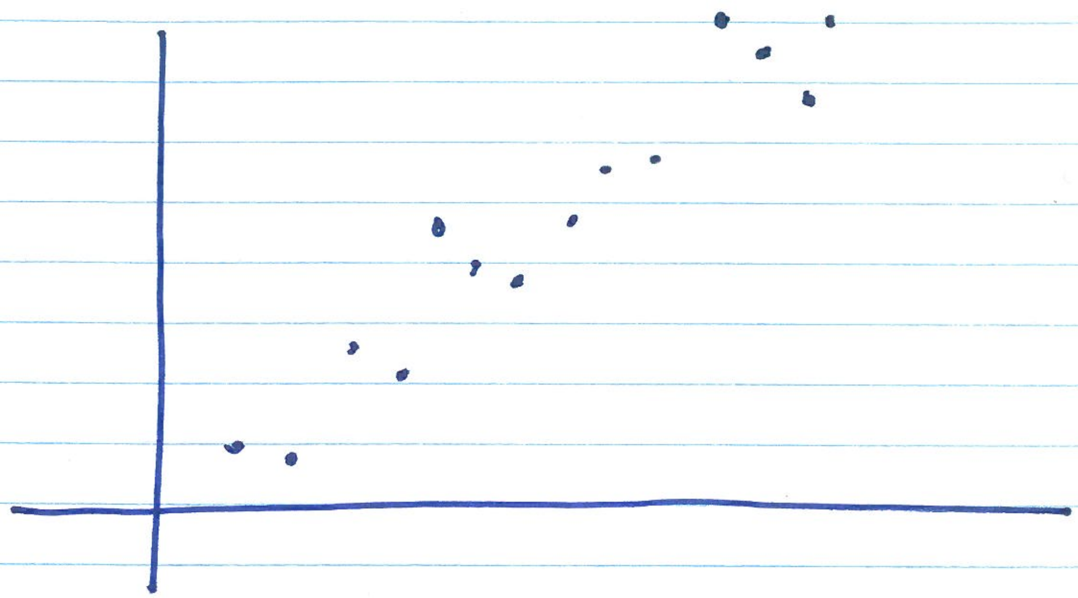
In many applications it is necessary to approximate a complicated or expensive to evaluate function, or a function only known at a discrete set of points, by a "simple" function which can easily be evaluated over a whole interval.

In cases where the function to be approximated is known accurately at a set of points, one is inclined to use an <u>interpolation</u> procedure — the graph of the approximating function runs exactly through the "data points".



If the data are expected to contain an error, which is the case for measurements or observations in experimental studies, a better strategy is to allow the graph of the approximating function to stray from the data points.

ex



A useful and well-known class of functions for mapping the set of real numbers into itself is the class of algebraic polynomials

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$

where n is a non-negative integer and a_i are real constants.

Polynomials have the desirable property that they can approximate any function over a closed, bounded interval.

This is precisely captured by the ...

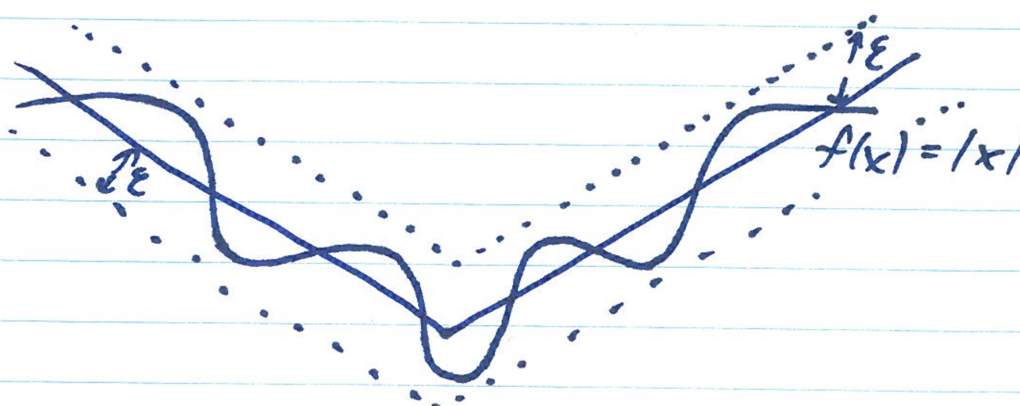
Weierstrass Approximation Theorem.

Suppose f is defined and continuous on $[a, b]$.

For each $\varepsilon > 0$ there exists a polynomial $P(x)$ defined on $[a, b]$ with the property that

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

for all x in $[a, b]$.



polynomial approximation to $f(x) = |x|$.

This is a very strong theorem — it only requires $f(x)$ to be continuous on the interval → it does not have to be differentiable.

Unfortunately, the Weierstrass Approximation Thm does not tell us how to select such a polynomial.

You might expect that Taylor series polynomials would be prominent in polynomial interpolation

Recall: $P(x) = f(a) + f'(a)(x-a) + \frac{1}{2}f''(a)(x-a)^2 + \cdots + \frac{1}{n!}f^{(n)}(a)(x-a)^n$

is an n^{th} degree polynomial that approximates f near "a."

Unfortunately, Taylor series concentrates its accuracy at the point "a" rather than over the entire interval, and are typically poorly suited for interpolation.

A particularly clear demonstration of this fact is seen for

$$f(x) = \frac{1}{x} \quad \text{expanded about } x_0 = 1.$$

Then $P_n(x) = \sum_{k=0}^{n} \frac{f^{(k)}(1)}{k!}(x-1)^k$

$$= \sum_{k=0}^{n} (-1)^k (x-1)^k$$

To approximate $f(3) = 1/3$ by $P_n(3)$ for increasing values of n gives a dramatic failure...

n	0	1	2	3	4	5	6	7
$P_n(3)$	1	-1	3	-5	11	-21	43	-85

We now focus on methods which use information throughout the interval to approximate f.

Polynomial Interpolation

We now assume that the given data are exact and represent values of some unknown function.

We want to find the polynomial $P(x)$ of smallest possible degree such that

$$P(x_k) = f_k \qquad k = 0, \dots, N$$

for $N+1$ distinct interpolation points $x_0, \dots, x_N$ and $N+1$ values $\underbrace{f_0, \dots, f_N}_{\text{data points}}$.

To solve this problem, we will first investigate the simpler problem where all the data equals zero, except at one point.

11.7

We are looking for a polynomial $L_m(x)$ of degree $\leq N$ such that

$$L_m(x_k) = \delta_{mk} = \begin{cases} 1 & k = m \\ 0 & k \neq m \end{cases}$$

This is known as the Kronecker Delta

This is easy to find.

Since the polynomial must vanish at the points x_k, $k \neq m$ it must contain the factors

$$(x - x_k) \quad \text{for } k \neq m$$

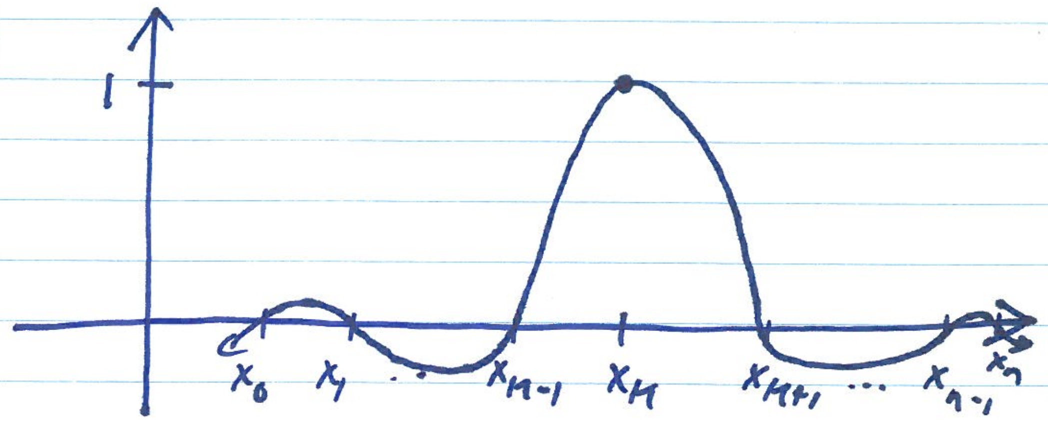
$$\therefore \quad L_m(x) = \text{Const} \times \prod_{\substack{k=0 \\ k \neq m}}^{N} (x - x_k)$$

The constant is determined by the condition $L_m(x_m) = 1$

$$\Rightarrow L_m(x) = \prod_{\substack{k=0 \\ k \neq m}}^{N} \frac{x - x_k}{x_m - x_k}$$

Typically,

$L_M(x)$



These polynomials are the building blocks for deriving a polynomial interpolating a general function

It is easy to see that

$$P(x) = \sum_{m=0}^{N} f_m L_m(x)$$

is a polynomial of degree $\leq N$ satisfying the interpolation conditions.

This polynomial is called the n^{th} Lagrange interpolating polynomial.

Recall that we wanted to find the polynomial $P(x)$ of smallest possible degree such that

$$P(x_k) = f_k \qquad k = 0, \ldots, N$$

Q. Is this interpolating polynomial unique?

A. Assume there are two different polynomials p and q of degree $\leq N$, which both satisfy the interpolation conditions.

$d = p - q$

Their difference is a polynomial of degree $\leq N$ and vanishes at the $N+1$ distinct points $x_0, \ldots, x_N$.

However a nonzero polynomial of degree $\leq N$ has at most N zeros, thus

$$d = p - q \equiv 0 \quad \Rightarrow p = q$$
$$\Rightarrow \text{Uniqueness.}$$

$\therefore$ The n^{th} Lagrange interpolating polynomial is the unique interpolating polynomial satisfying the interpolation conditions.

Ex: Fit a cubic through the first four points of the table

i	x_i	$f(x_i)$
0	3.2	22.0
1	2.7	17.8
2	1.0	14.2
3	4.8	38.3
4	5.6	51.7

and use it to find the interpolated value for $x = 3.0$.

Soln: The 3^{rd} Lagrange interpolating polynomial is given by

$$P(x) = \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} f_0$$

$$+ \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)} f_1$$

$$+ \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} f_2$$

$$+ \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)} f_3$$

Substituting the values from the table and evaluating at $x = 3$ gives

$$P(3.0) = \frac{(3.0-2.7)(3.0-1.0)(3.0-4.8)}{(3.2-2.7)(3.2-1.0)(3.2-4.8)}(22.0)$$
$$+ \frac{(3.0-3.2)(3.0-1.0)(3.0-4.8)}{(2.7-3.2)(2.7-1.0)(2.7-4.8)}(17.8)$$
$$+ \frac{(3.0-3.2)(3.0-2.7)(3.0-4.8)}{(1.0-3.2)(1.0-2.7)(1.0-4.8)}(14.2)$$
$$+ \frac{(3.0-3.2)(3.0-2.7)(3.0-1.0)}{(4.8-3.2)(4.8-2.7)(4.8-1.0)}(38.3)$$

$$= 20.21$$

Our next task is to develop estimates for the error. As it turns out, the form of the error (but not necessarily the magnitude) resembles that for the n^{th} Taylor Polynomial.

<u>Thm</u>: (3.3 of text)

Suppose $x_0, x_1, \dots, x_n$ are distinct numbers in the interval $[a, b]$ and $f \in C^{n+1}[a,b]$. Then for each x in $[a, b]$ a number $\xi(x)$ in (a, b) exists with

$$f(x) = P(x) + \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x-x_0)(x-x_1)\cdots(x-x_n)$$

n^{th} Lagrange interpolating polynomial.

Recall from last day:

If f has $(n+1)$ continuous derivatives on $[a, b]$ &
$P(x)$ is the interpolating polynomial of degree $\leq n$ interpolating f at the points $x_0, x_1, \ldots x_n$
then

$$f(x) - P(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \prod_{k=0}^{n} (x - x_k)$$

$\xi \in (a, b)$

<u>Ex</u>. Suppose you need to construct six-decimal-place tables for the common, or base-10, logarithm function from $x = 1$ to $x = 10$ in such a way that linear interpolation is accurate to within 10^{-6}. Determine a bound for the step size for this table.

Ex Based on the following data

i	x_i	$f(x_i)$
0	3.2	22.0
1	2.7	17.8
2	1.0	14.2
3	4.8	38.3
4	5.6	51.7

find approximations to $f(3)$ using the 2^{nd} and 3^{rd} Lagrange interpolating polynomials.

Solns We will use x_0, x_1 and x_3 to build the 2^{nd} Lagrange interpolating polynomial

$$P_2(3) = \frac{(3-2.7)(3-4.8)}{(3.2-2.7)(3.2-4.8)}(22.0)$$

$$+ \frac{(3-3.2)(3-4.8)}{(2.7-3.2)(2.7-4.8)}(17.8)$$

$$+ \frac{(3-3.2)(3-2.7)}{(4.8-3.2)(4.8-2.7)}(38.3)$$

$$\approx 20.27$$

We will use x_0, x_1, x_2, x_3 to build the 3^{rd} Lagrange interpolating polynomial:

$$P_3(3) = \frac{(3.0-2.7)(3.0-1.0)(3.0-4.8)}{(3.2-2.7)(3.2-1.0)(3.2-4.8)}(22.0)$$

$$+ \frac{(3.0-3.2)(3.0-1.0)(3.0-4.8)}{(2.7-3.2)(2.7-1.0)(2.7-4.8)}(17.8)$$

$$+ \frac{(3.0-3.2)(3.0-2.7)(3.0-4.8)}{(1.0-3.2)(1.0-2.7)(1.0-4.8)}(14.2)$$

$$+ \frac{(3.0-3.2)(3.0-2.7)(3.0-1.0)}{(4.8-3.2)(4.8-2.7)(4.8-1.0)}(38.3)$$

$$\approx 20.21$$

Notice that: 1 The error formula was not applied because we do not know the derivative values of f. But we can get an estimate for the error by examining polynomials of different degree & by using different nodes.

2 The P_2-calculation was not used to reduce the work in calculating P_3. We want a way to use previous values, especially since point 1 means we will examine results for polynomials of varying degree.

We want to examine polynomials based on different nodes.

In the last example, we considered the polynomial based on the nodes x_0, x_1 and x_3.

Call this polynomial $P_{013}(x)$.

We also considered the polynomial based on nodes x_0, x_1, x_2, x_3.

Call this polynomial $P_{0123}(x)$.

Similarly, we make the following definition:

Let f be a function defined at $x_0, x_1, x_2, \ldots, x_n$ and suppose that $m_1, m_2, \ldots, m_K$ are K distinct integers with $0 \leq m_i \leq n$ for each i.

The Lagrange polynomial that agrees with f at $x_{m_1}, x_{m_2}, \ldots, x_{m_K}$ is denoted $P_{m_1, m_2, \ldots, m_K}$

Using this notation,

$$P_0(x) =$$

$$P_1(x) =$$

$$P_{0,1}(x) = \frac{(x-x_1)}{(x_0-x_1)} f(x_0) + \frac{(x-x_0)}{(x_1-x_0)} f(x_1)$$

$$= \frac{(x-x_1)P_0(x) - (x-x_0)P_1(x)}{(x_0-x_1)}$$

So $P_{0,1}(x)$ can be recursively defined in terms of P_0 and P_1

More generally

<u>Thm</u>: Let f be defined at $x_0, x_1, \ldots, x_k$ and x_j and x_i be two distinct numbers in this set. Then

$$P_{0,1,\ldots,k}(x)$$

$$= \frac{(x-x_j)P_{0,1,\ldots,j-1,j+1,\ldots,k}(x) - (x-x_i)P_{0,1,\ldots,i-1,i+1,\ldots,k}(x)}{(x_i-x_j)}$$

Proof: Let $\hat{Q}(x) = P_{0,1,\dots,j-1,j+1,\dots,k}(x)$

$$Q(x) = P_{0,1,\dots,i-1,i+1,\dots,k}(x)$$

$$P(x) = \frac{(x-x_j)\hat{Q}(x) - (x-x_i)Q(x)}{(x_i - x_j)}$$

If $r \neq i$, $r \neq j$ then

$$Q(x_r) = \hat{Q}(x_r) = f(x_r) \qquad 0 \le r \le k$$

$$\therefore P(x_r) = \frac{(x_r - x_j)\hat{Q}(x_r) - (x_r - x_i)Q(x_r)}{(x_i - x_j)}$$

$$= \frac{(x_i - x_j)}{(x_i - x_j)} f(x_r) = f(x_r)$$

Also $P(x_i) = \dfrac{(x_i - x_j)\hat{Q}(x_i) - (x_i - x_i)Q(x_i)}{(x_i - x_j)}$

$$= \hat{Q}(x_i) = f(x_i)$$

Similarly $P(x_j) = f(x_j)$.

By definition, $P_{0,1,\dots,k}(x)$ is the unique polynomial of degree at most k which agrees with f at $x_0, \dots, x_k$.

$$\Longrightarrow P = P_{0,1,\dots,k}(x)$$

The corresponding procedure is called Neville's Method. Here, values for each interpolating polynomial are generated using previous calculations.

E.g., $P_{0,1}(x) = \dfrac{(x-x_1)P_0 - (x-x_0)P_1}{(x_0-x_1)}$

is derived from P_0 & P_1

$P_{1,2}(x)$ is derived from P_1 & P_2

$\vdots$

Written as a table:

$$
\begin{tikzcd}
P_0 \arrow[dr] & & & & \\
& P_{0,1} \arrow[dr] & & & \\
P_1 \arrow[ur] \arrow[dr] & & P_{0,1,2} \arrow[dr] & & \\
& P_{1,2} \arrow[ur] \arrow[dr] & & P_{0,1,2,3} \arrow[dr] & \\
P_2 \arrow[ur] \arrow[dr] & & P_{1,2,3} \arrow[ur] \arrow[dr] & & P_{0,1,2,3,4} \\
& P_{2,3} \arrow[ur] \arrow[dr] & & P_{1,2,3,4} \arrow[ur] & \\
P_3 \arrow[ur] \arrow[dr] & & P_{2,3,4} \arrow[ur] & & \\
& P_{3,4} \arrow[ur] & & & \\
P_4 \arrow[ur] & & & &
\end{tikzcd}
$$

Ex. Suppose $x_j = j$ for $j = 0, 1, 2, 3$ and it is known that

$$P_{0,1}(x) = 2x + 1$$

$$P_{0,2}(x) = x + 1$$

$$P_{1,2,3}(2.5) = 3$$

Find $P_{0,1,2,3}(2.5)$.

Notice that

- With Neville's Method, if the latest approximation is not as accurate as desired, another node can be selected, and another row added to the table.
- Unfortunately, if we subtract a point from the set used to construct the polynomial, we essentially have to start over in the computations.
- Neville's Method also has to repeat all the arithmetic if we interpolate at a new x-value.

<u>Divided Differences</u> avoid all of this extra computation. This approach also uses fewer arithmetic operations than Neville's Method.

Divided Differences.

To develop divided differences, we will assume that the function $f(x)$ is known at several values for x:

$$\begin{array}{ccccc} f_0 & f_1 & f_2 & f_3 & \dots \\ x_0 & x_1 & x_2 & x_3 & \end{array}$$

We do not assume that the x's are evenly spaced or even that the values are arranged in any particular order.

Consider the n^{th} degree polynomial written in a special way

$$P_n(x) = a_0 + (x-x_0)a_1 + (x-x_0)(x-x_1)a_2 + \dots + (x-x_0)(x-x_1)\cdots(x-x_{n-1})a_n$$

If we chose the a_i so that $P_n(x) = f(x)$ at the $n+1$ known points then $P_n(x)$ is an interpolating polynomial.

We will show that the a_i are readily determined by using what are called the divided differences.

First determine a_0: $P_n(x_0) = f(x_0) = a_0$

and define the zeroth divided difference of the function f:

$$f[x_i] \equiv f(x_i)$$

Now determine a_1:

$$P_n(x_1) = f(x_1) = a_0 + (x_1 - x_0)\, a_1$$

$$\therefore \quad a_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{f[x_1] - f[x_0]}{x_1 - x_0}$$

and define the first divided difference of f with respect to x_i and x_{i+1}

$$f[x_i, x_{i+1}] = f[x_{i+1}, x_i] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i}$$

Similarly, the second divided difference is defined as

$$f[x_i, x_{i+1}, x_{i+2}] = \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i}$$

and we define the k^{th} divided difference relative to

$$x_i, x_{i+1}, x_{i+2}, \ldots, x_{i+k}$$

as

$$f[x_i, x_{i+1}, \ldots, x_{i+k}] = \frac{f[x_{i+1}, x_{i+2}, \ldots, x_{i+k}] - f[x_i, x_{i+1}, \ldots, x_{i+k-1}]}{x_{i+k} - x_i}$$

In a similar fashion to the evaluation of a_0 and a_1 we can show

$$a_k = f[x_0, x_1, \ldots, x_k]$$

This gives Newton's interpolatory divided difference formula.

$$\begin{aligned} P_n(x) = {} & f[x_0] + f[x_0, x_1](x - x_0) \\ & + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\ & + \\ & \vdots \\ & + \\ & f[x_0, x_1, \ldots, x_n](x - x_0)(x - x_1) \cdots (x - x_{k-1}) \end{aligned}$$

Ex It is of interest to look at divided differences for a polynomial. Suppose that $f(x)$ is the cubic

$$f(x) = 2x^3 - x^2 + x - 1$$

Here is the divided difference table

x_i	$f[x_i]$	$f[x_{i-1}, x_i]$	$f[x_{i-2},\ldots,x_i]$	$f[x_{i-3},\ldots x_i]$	$f[x_{i-4},\ldots x_i]$
0.30	-0.7360				
		2.4800			
1.00	1.0000		3.0000		
		3.6800		2.0000	
0.70	-.1040		3.6000		0
		2.2400		2.0000	
0.60	-.3280		5.4000		
		8.7200			
1.90	11.0080				

Notice that the third divided differences are all the same. It then follows that all higher divided differences will be zero.

Based on the table

$$f(x) = \_\_\_\_\_\_ + \_\_\_\_\_\_(x-0.30)$$

$$+ \_\_\_\_\_\_(x-.30)(x-1) + \_\_\_\_\_\_(x-.3)(x-1)(x-.7)$$

13.5

If the x-values are evenly spaced, getting an interpolating polynomial is simplified. Instead of using divided differences, "ordinary differences" are used — the differences in f values are not divided by the differences in x-values.

A delta symbol is used to write them:

$$\Delta f_i = \underbrace{f_{i+1} - f_i}_{\text{forward difference.}}$$

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

$$\vdots$$

Now let $x = x_0 + sh \Rightarrow x - x_i = (s-i)h$

& $$P(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \cdots + f[x_0, x_1, \ldots, x_n](x - x_0)\cdots(x - x_{n-1})$$

$$\Rightarrow P(x) = f(x_0) + s\Delta f_0 + \frac{s(s-1)}{2!}\Delta^2 f_0 + \cdots + \frac{s(s-1)(s-2)\cdots(s-n+1)}{n!}\Delta^n f_0$$

Using binomial coefficient notation

$$\binom{s}{k} = \frac{s(s-1)(s-2)\cdots(s-k+1)}{k!}$$

$$\Rightarrow \quad p(x) = \sum_{k=0}^{n} \binom{s}{k} \Delta^k f_0$$

This is known as the Newton Forward Difference Formula.

An alternative approach based on backward differences is also possible

Let $\quad \nabla p_n = \underbrace{p_n - p_{n-1}}_{\text{backwards difference}} \quad \left(\begin{array}{c}\text{read nabla} \\ p_n\end{array}\right)$

$$\nabla^2 p_n = \nabla(\nabla p_n) = p_n - 2p_{n-1} + p_{n-2}$$

$$\vdots$$

Then

$$f[x_n, x_{n-1}] = \frac{1}{h} \nabla f_n$$

$$f[x_n, x_{n-1}, x_{n-2}] = \frac{1}{2h^2} \nabla^2 f_n$$

$$\vdots$$

$$f[x_n, x_{n-1}, \ldots, x_{n-k}] = \frac{1}{k!\,h^k} \nabla^k f_n$$

Notice that

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

Implies that

$$P(x) = f[x_n] + f[x_n, x_{n-1}](x - x_n) + f[x_n, x_{n-1}, x_{n-2}](x - x_n)(x - x_{n-1}) + \cdots + f[x_n, x_{n-1}, \ldots, x_0](x - x_n)\cdots(x - x_1)$$

Why? (reorder interpolating nodes as $x_n, x_{n-1}, \ldots, x_0$)

Thus

$$P(x) = f_n + s\nabla f_n + \frac{s(s+1)}{2}\nabla^2 f_n + \cdots + \frac{s(s+1)\cdots(s+n-1)}{n!}\nabla^n f_n$$

which is Newton's Backward-Difference Formula.

Which should we use?
Newton's Forward Difference Formula
or Newton's Backward Difference Formula?
to evaluate $P(x)$?

If x is close to x_0, then the Forward Difference Formula is preferred because we want to make the earliest use of data points near x (helps to avoid cancellation error)

Similarly, Newton's Backward Difference Formula is prefered if x is close to x_n.

There are also divided difference formulas that are preferred when x lies near the center of the table (these are Centered-Difference Formulas)

An example of a centered difference formula is Stirling's formula. See text.

Suppose that we use the n^{th} Lagrange polynomial to approximate a function f.

$P(x)$ agrees with f at the nodes $x_0, x_1, \ldots, x_n$.

But $P(x)$ is an n^{th} degree polynomial!

Problems

- $P(x)$ is expensive to evaluate if n is large.
- ** The interpolating polynomial can oscillate wildly, except where contained between data points that are in close proximity. (see Fig 3.12 of text).

Q The following data are given for a polynomial $P(x)$ of unknown degree.

x	0	1	2
$P(x)$	2	-1	4

Determine the coefficient of x^2 in $P(x)$ if all third order forward differences are 1.