Example Compute div. diff table for the following function values.

x	3	1	5	6
$f(x)$	1	-3	2	4

$$\begin{array}{cc|cccc} 3 & 1 & 2 & -3/8 & 7/40 \\ 1 & -3 & 5/4 & 3/20 & \\ 5 & 2 & 2 & & \\ 6 & 4 & & & \end{array}$$

Thus the Newton interp poly for f at the given nodes is:

$$p(x) = 1 + 2(x-3) - \frac{3}{8}(x-3)(x-1) + \frac{7}{40}(x-3)(x-1)(x-5)$$

(we took values from top most row of table)

Divided differences Algorithm

Let us use the following notation for the table entries:

row column

$$\begin{array}{cc|cccccc} x_0 & c_{00} & c_{01} & c_{02} & c_{03} & \cdots & c_{0n-1} & c_{0n} \\ x_1 & c_{10} & c_{11} & c_{12} & c_{13} & \cdots & c_{1n-1} & \\ x_2 & c_{20} & c_{21} & c_{22} & c_{23} & & & \\ \vdots & & \vdots & \vdots & & & & \\ x_{n-1} & c_{n-1 0} & c_{n-1 1} & & & & & \\ x_n & c_{n0} & & & & & & \end{array}$$

i.e. $\boxed{c_{ij} = f[x_i, x_{i+1}, \ldots, x_{i+j}]}$

Algorithm:

for $j = 1 \ldots n$

  for $i = 0 \ldots n-j$

    $c_{ij} = (c_{i+1, j-1} - c_{i\, j-1}) / (x_{i+j} - x_i)$

Here we need to store all c_{ij}, however the algorithm can be easily modified to do all modifications in place.

for $i = 0 .. n$
  $d_i = f(x_i)$

initialization identical to usual table.

for $j = 1 .. n$
  for $i = n:-1:j$
    $d_i = (d_i - d_{i-1})/(x_i - x_{i-j})$

Sample run:

init	$j=1$	$j=2$	$j=3$	$j=4$
x	x	x	x	x
•	x	x	x	x
•	•	x	x	x
•	•	•	x	x
•	•	•	•	x

• = same value

x = final value

note: inner loop works its way upward so as to leave top part of vector unchanged (this is why we loop $n:-1:j$.)

At end of algorithm the interp. poly is:

$$p(x) = \sum_{i=0}^{n} d_i \prod_{j=0}^{i-1} (x - x_j)$$

<u>Theorem (on permutation of div. diff)</u>

Let $z_0, z_1, \dots, z_n$ be a permutation of $x_0, x_1, \dots, x_n$

then

$$f[z_0, z_1, \dots, z_n] = f[x_0, x_1, \dots, x_n]$$

in words: divided diff. remain unchanged after perm of arguments.

<u>proof</u>: $f[z_0, z_1, \dots, z_n] =$ coeff in front of x^n of interp poly of f at pts $z_0, z_1, \dots z_n$

$=$ coeff in front of x^n of interp poly of f at pts $x_0, x_1, \dots, x_n$

$= f[x_0, x_1, \dots, x_n]$.

We now give <u>explicit</u> formula for interp error

<u>Theorem (on error of Newton interp.)</u>

Let p be the interp poly of f at the $n+1$ distinct points $x_0, x_1, \dots x_n$. If t is different from nodes:

$$f(t) - p(t) = f[x_0, x_1, \dots, x_n, t] \prod_{j=0}^{n} (t - x_j)$$

<u>proof</u>: Let $q \equiv$ inter poly of degree $\leq n+1$ interp f at $n+2$ points $x_0, x_1, \dots, x_n, t$.

By construction and since p interpolates f at $x_0, \dots, x_n$:

$$q(x) = p(x) + f[x_0, x_1, \dots, x_n, t] \prod_{j=0}^{n} (x - x_j)$$

eval at $x = t$:

$$f(t) = q(t) = p(t) + f[x_0, x_1, \dots x_n, t] \prod_{j=0}^{n} (t - x_j)$$

$\rightarrow$ which gives result.

Theorem derivatives and divided differences

If $f \in C^n[a,b]$ and if $x_0, \dots, x_n$ are $n+1$ distinct points in $[a,b]$ $\exists\, \xi \in (a,b)$ s.t.

$$f[x_0, x_1, x_2, \dots, x_n] = \frac{1}{n!} f^{(n)}(\xi)$$

proof: This is a combination of

- interp error theorem
- ——— in Newton form.

indeed; interp error theorem says that if p interpolates f at the nodes $x_0, x_1, \dots, x_{n-1}$ then $\exists\, \xi \in (a,b)$ s.t.

$$f(x_n) - p(x_n) = \frac{1}{n!} f^{(n)}(\xi) \prod_{j=0}^{n-1} (x - x_j)$$

moreover previous theorem $= f[x_0, x_1, \dots, x_n] \prod_{j=0}^{n-1} (x - x_j)$

$\rightsquigarrow$ which gives result.

§3.3 Hermite interpolation

Question: can we find a polynomial matching a function and some of its derivatives at a set of nodes?

generally speaking Lagrange interp. ≡ only function values

Hermite interp. ≡ function values + derivatives,

Here is an example of how to proceed.

Problem: find polynomial p s.t.

$$p(x_0) = f(x_0) \qquad p'(x_0) = f'(x_0)$$
$$p(x_1) = f(x_1) \qquad p'(x_1) = f'(x_1)$$