Divided Differences & Interpolation/Extrapolation

Here is the divided difference table for f based on x_0, x_1, x_2, x_3, x_4.

x_i	$f[\,] = f(\,)$	$f[,]$	$f[,,]$	$f[,,,]$	$f[,,,,]$
x_0	$f(x_0)$	$f[x_0, x_1]$	$f[x_0, x_1, x_2]$	$f[x_0, x_1, x_2, x_3]$	$f[x_0, x_1, x_2, x_3, x_4]$
x_1	$f(x_1)$	$f[x_1, x_2]$	$f[x_1, x_2, x_3]$	$f[x_1, x_2, x_3, x_4]$	
x_2	$f(x_2)$	$f[x_2, x_3]$	$f[x_2, x_3, x_4]$		
x_3	$f(x_3)$	$f[x_3, x_4]$			
x_4	$f(x_4)$				

Here $f[x_i] = f(x_i), f[x_i, x_{i+1}] = \dfrac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i}$, and generally, $f[x_i, \ldots, x_{i+k}] = \dfrac{f[x_{i+1}, \ldots, x_{i+k}] - f[x_i, \ldots, x_{i+k-1}]}{x_{i+k} - x_i}$. Remarkably, this depends only on the values $x_i, \ldots, x_{i+k}$—not their order.

The interpolating polynomial based on the points $x_0, x_1, \ldots, x_n$ and which corresponds to the Newton advancing difference formula may be constructed by inserting a row at the top, thus:

x_i	$f[\,] = f(\,)$	$f[,]$	$f[,,]$	$f[,,,]$	$f[,,,,]$
x	$f(x)$	$f[x, x_0]$	$f[x, x_0, x_1]$	$f[x, x_0, x_1, x_2]$	$f[x, x_0, x_1, x_2, x_3]$
x_0	$f(x_0)$	$f[x_0, x_1]$	$f[x_0, x_1, x_2]$	$f[x_0, x_1, x_2, x_3]$	$f[x_0, x_1, x_2, x_3, x_4]$
x_1	$f(x_1)$	$f[x_1, x_2]$	$f[x_1, x_2, x_3]$	$f[x_1, x_2, x_3, x_4]$	
x_2	$f(x_2)$	$f[x_2, x_3]$	$f[x_2, x_3, x_4]$		
x_3	$f(x_3)$	$f[x_3, x_4]$			
x_4	$f(x_4)$				

The "$f[x, \ldots]$" divided differences satisfy

$f[x, x_0, \ldots, x_{i-1}]$	$f[x, x_0, \ldots x_i] = \frac{f[x, x_0, \ldots, x_{i-1}] - f[x_0, x_1, \ldots, x_i]}{x - x_i}$
$f[x_0, x_1, \ldots, x_i]$	

so for i from n down to 1,

$$f[x, x_0, \ldots, x_{i-1}] = f[x_0, x_1, \ldots x_i] + (x - x_i) f[x, x_0, \ldots x_i];$$

for $i = 0$, $f[x] = f[x_0] + (x - x_0) f[x, x_0]$. Backtracking we get [2]

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

This formula uses the divided differences in a row (in this case, the top row) of the table. In a practical application, this value would be calculated assuming that the final term, the "error term," is zero.

The equivalent interpolating formula can also be calculated from the divided differences that lie on the bottom diagonal. These divided differences satisfy

$f[x_{i-1}, x_i, \dots, x_n]$	$f[x_{i-1}, \dots, x_n, x] = \frac{f[x_i,\dots,x_n,x]-f[x_{i-1},x_i,\dots,x_n]}{x-x_{i-1}}$
$f[x_i, \dots, x_n, x]$	

so for i from 1 to n,

$$f[x_i, \dots, x_n, x] = f[x_{i-1}, x_i, \dots, x_n] + (x - x_{i-1})f[x_{i-1}, \dots, x_n, x].$$

Then $f(x) = f(x_n) + (x - x_n)f[x_n, x]$.

Here is an alternative explanation of the second way of deriving the interpolating polynomial based on the reference [1] by Conte and de Boor, whose notation has also been used here. The Newton interpolating polynomial based on the points $x_n, x_{n-1}, \dots, x_0$ is given by

$$p_n(x) = \sum_{i=0}^{n} f[x_i, \dots, x_n] \prod_{j=i+1}^{n} (x - x_j).$$

Here $\prod_{j=i+1}^{n} (x - x_j)$ denotes the product of all factors $(x - x_j)$ for $i + 1 \le j \le n$. (An "empty" product is 1.) The calculation may be done by the algorithm:

```
Set $v_0 = f[x_0, \dots, x_n]$;
for $i = 1, \dots, n$ do:
set $v_i = f[x_i, \dots, x_n] + (x - x_i)v_{i-1}$.
```

The result is $v_n = p_n(x)$. This is the interpolating polynomial of degree $\le n$: $p_n(x_0) = f(x_0), p_n(x_1) = f(x_1), \dots, p_n(x_n) = f(x_n)$.

You can check that applying this algorithm with $n = 3$ and $x = x_4$ is equivalent to reconstructing $f(x_4)$ from the following table (actually we use just the last values in each column).

x_i	$f[\,] = f(\,)$	$f[,]$	$f[,,]$	$f[,,,]$	$f[,,,,]$
x_0	$f(x_0)$	$f[x_0, x_1]$	$f[x_0, x_1, x_2]$	$f[x_0, x_1, x_2, x_3]$	$f[x_0, x_1, x_2, x_3, x_4] = 0$
x_1	$f(x_1)$	$f[x_1, x_2]$	$f[x_1, x_2, x_3]$		
x_2	$f(x_2)$	$f[x_2, x_3]$			
x_3	$f(x_3)$				
x_4					

Thus, this algorithm is equivalent to taking $x_{n+1} = x$, assuming $f[x_0, x_1, \dots, x_n, x_{n+1}] = 0$, and using the divided differences to reconstruct $f[x_{n+1}]$. Then $p_n(x) = f[x_{n+1}]$.

REFERENCES

[1] S. D. Conte and Carl de Boor, *Elementary Numerical Analysis: an Algorithmic Approach* 2/e, McGraw-Hill, New York, 1972, pp. 202–203.

[2] Stephen G. Kellison, *Fundamentals of Numerical Analysis*, Richard D. Irwin, Inc., Homewood, Illinois, 1975, p. 108.