A simple algorithm can be given for constructing the differences

$$f(x_0), f[x_0, x_1], f[x_0, x_1, x_2], \ldots, f[x_0, x_1, \ldots, x_n]$$

which are necessary for evaluating the Newton form (3.2.9).

Algorithm *Divdif* (d, x, n)

1. Remark: d and x are vectors with entries $f(x_i)$ and x_i, $i = 0, 1, \ldots, n$, respectively. On exit, d_i will contain $f[x_0, \ldots, x_i]$.

2. Do through step 4 for $i = 1, 2, \ldots, n$.

3. Do through step 4 for $j = n, n - 1, \ldots, i$.

4. $d_j := (d_j - d_{j-1})/(x_j - x_{j-i})$.

5. Exit from the algorithm.

To evaluate the Newton form of the interpolating polynomial (3.2.9), we give a simple variant of the nested polynomial multiplication (2.9.8) of Chapter 2.

Algorithm Interp (d, x, n, t, p)

1. Remark: On entrance, d and x are vectors containing $f[x_0, \ldots, x_i]$ and x_i, $i = 0, 1, \ldots, n$, respectively. On exit, p will contain the value $p_n(t)$ of the nth-degree polynomial interpolating f on x.

2. $p := d_n$

3. Do through step 4 for $i = n - 1, n - 2, \ldots, 0$.

4. $p := d_i + (t - x_i)p$

5. Exit the algorithm.