ALGORITHM 70
INTERPOLATION BY AITKEN

CHARLES J. MIFSUD
General Electric Co., Bethesda, Md.

procedure AITKEN (x, f, n, X, F);
    real array x, f;
    integer n;    real X, F;

comment If given \(x_0, x_1, \ldots, x_n\), \(n+1\) abscissas and also given \(f(x_0), f(x_1), \ldots, f(x_n)\), \(n+1\) functional values, this procedure generates a Lagrange polynomial, \(F(X)\) of the \(n\)th degree so that \(F(x_k) = f(x_k)\). Hence, for any given value \(X\), a functional value \(F(X)\) is generated. The procedure is good for either equal or unequal intervals of \(x_i\). Aitken's iterative scheme is used in the generation of \(F(X)\). Since the \(f\) array is used for temporary storage, as the calculation proceeds its original values are destroyed;

begin integer i, j, t;
    for j := 0 step 1 until n-1 do
        begin t := j+1
            for i := t step 1 until n do
                f[i] := \((X-x_{ij}) \times f[i] - (X-x_{ij}) \times f[j]/(x[i]-x[j])\)
        end
        F := f[n]
end

CERTIFICATION OF ALGORITHM 70
INTERPOLATION BY AITKEN [C. J. Mifsud, Comm. ACM 4 (Nov. 1961)]
A. P. RELPH
The English Electric Co., Whetstone, England

Algorithm 70 was translated using the DEUCE ALGOL compiler and gave satisfactory results after semicolons had been added to

\[ t := j+1 \text{ to make it } t := j+1; \]

and \( (x[i]-x[j]) \text{ end to make it } (x[i]-x[j]) \text{ end}; \)

The identifier \(t\) can be eliminated and the algorithm shortened by the following changes:

\textbf{Replace} \hspace{2em} \textbf{begin integer} i, j, t; \hspace{2em} \textbf{by} \hspace{2em} \textbf{begin integer} i, j;

\textbf{Replace} \hspace{2em} t := j+1; \hspace{2em} \textbf{by} \hspace{2em} \textbf{for} i := j+1 \textbf{ step 1 until}

\textbf{for} i := t \textbf{ step 1 until n do}