## 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 x0 ,x1 , . . . xn ,  n+1 abscissas and also given
  f(x0), f(x1), . . . f(xn),  n+1 functional values, this procedure
  generates a Lagrange polynomial, F(X) of the nth degree so that
  F(xi) = f(xi). Hence, for any given value X, a functional value
  F(X) is generated. The procedure is good for either equal or
  unequal intervals of the xi. Aitken's interative scheme is used
  in the generation of F(X). Since the f array is used for tem-
  porary 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 [j]) × f [i] - (X-x [i]) × 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 *to make it* t := j+1;

and (x[i]-x[j]) **end** *to make it* (x[i]-x[j]) **end**;

The identifier *t* can be eliminated and the algorithm shortened by the following changes:

*Replace* **begin integer** i, j, t; *by* **begin integer** i, j;

*Replace* t := j+1; **for** i := t **step** 1 **until** n **do** *by* **for** i := j+1 **step** 1 **until** n **do**