ALGORITHM 79
DIFFERENCE EXPRESSION COEFFICIENTS

THOMAS P. GIAMMO
Space Technology Laboratories, Inc., Los Angeles, California

```
procedure dicol (k, n, xp, xtab, coef);
value k, n; integer k, n; real xp;
array xtab, coef;
comment dicol produces the coefficients for the n ordinates
  (corresponding to the abscissae, xtab) in the n-point finite
  difference expression for the k-th derivative evaluated at xp.
  The method used is to determine the analytic expression for
  the k-th derivative of each coefficient in the n-point Lagrangian
  interpolation formula and evaluate it at xp. Note that k=0
  will produce the Lagrangian interpolation coefficients them-
  selves;
begin integer array xuse [1 : n-1]; real factk, sum, denom,
  part;
integer i, terms, j, m, high;
factk := 1.0; for i := 2 step 1 until k do factk := i×factk;
terms := n-k-1; if terms<0 then go to Z;
for j := 1 step 1 until n do
loop: begin sum := 0; denom := 1.0; part := 1.0;
        for i := 1 step 1 until n do
        if i ≠ j then denom := denom×(xtab [j] - xtab [i]);
        if terms = 0 then go to Y;
        m := 1; high := 1;
    A:  if (high = j)∨(xtab [high] = xp) then
        A1: begin high := high + 1; go to A end A1;
        if high >.n then A2: begin m := m-1; if m>0
          then
        A3: begin high := xuse [m]+1; go to A end A3;
        go to X end A2;
        xuse [m] := high; m := m+1;
        if m≦terms then begin high := high + 1; go to
          A end;
        for i := 1 step 1 until terms do
          part := part×(xp - xtab [xuse [i]]);
        sum := sum + part; m := terms; part := 1.0;
        high := xuse [terms] + 1; go to A;
    Y:  sum := 1.0;
    X:  coef [j] := sum × factk/denom end loop;
      go to EXIT;
    Z:  for i := 1 step 1 until n do coef [i] := 0;
    EXIT: end dicol
```

CERTIFICATION OF ALGORITHM 79
DIFFERENCE EXPRESSION COEFFICIENTS

[Thomas Giamo, *Comm. ACM*, Feb. 1962]

EVA S. CLARK
University of California at San Diego, La Jolla, California

The procedure was translated into FORTRAN and run on the CDC 1604. Reasonable accuracy was obtained for $k = 0, 4 \leqq n \leqq 12$. For increasing n and increasing k, the accuracy diminished. It was found that the execution time increased rapidly as n was increased. For $k = 0$, the following results were obtained:

n	*Approximate Number of Machine Operations*
4	1.3×10^3
6	6.9×10^3
8	3.8×10^4
10	1.8×10^5
12	8.6×10^5

The author indicated in a letter that the procedure was developed for use with small n and small k.