```
   30 IF(W.LT.EPS) GO TO 130
      J=W/D
      V=FLOAT(J)*D
      U=W-V
      IF(U.LE.D1) GO TO 50
      IF(U.GT.(D-B2)) GO TO 40
      V=V+D1
      Z=B1+D2
      GO TO 60
   40 V=V+D
   50 V=V-B2
      Z=B2+D1
   60 U=(V-W)*(V+W)+V*Z
      IF(U.LT.0.0) V=V+Z
      W=W/V
      DL(1)=D1*W
      DL(2)=B1*W
      DL(3)=D2*W
      DL(4)=B2*W
      D=DL(1)
      J=1
      GO TO 80
   70 DL(1)=0.0

C
C     ENTERS PLOTTING SECTION.
C
      80 I=J
      90 IF(J.EQ.N) GO TO 130
         J=J+1
         IF(DL(1).LT.EPS) GO TO 120
         U=X(J)-X(I)
         V=Y(J)-Y(I)
         W=SQRT(U*U*XSS+V*V*YSS)
         IF(W.LT.EPS) GO TO 90
         DINTX=U/W
         DINTY=V/W
         W=W*W
C
C     CALCULATE INCREMENTS TO COMPLETE DASH OR BLANK NOT
C     FINISHED IN LAST JOIN.
C
         DX=DINTX*D
         DY=DINTY*D
     100 IF((DX*DX*XSS+DY*DY*YSS).GE.W) GO TO 110
         XX=X(I)+DX
         YY=Y(I)+DY
         CALL PENTO(L,XX,YY)
         L=3-L
         M=M+1
         IF(M.EQ.5) M=1
         DX=DX+DINTX*DL(M)
         DY=DY+DINTY*DL(M)
         GO TO 100
C
C     CALCULATES LENGTH OF UNFINISHED DASH OR BLANK.
C
     110 U=X(I)-X(J)+DX
         V=Y(I)-Y(J)+DY
         D=SQRT(U*U*XSS+V*V*YSS)
C
C     COMPLETES PRESENT JOIN.
C
     120 CALL PENTO(L,X(J),Y(J))
         GO TO 80
     130 CALL PENTO(2,X(N),Y(N))
     140 RETURN
         END
```

Algorithms supplement—Statement of Policy

A contribution to the Supplement may consist of an Algorithm, a Note on a previous algorithm, or an item under the heading of Correspondence.

Because the aim is to facilitate the interchange of algorithms, these should normally be submitted in one of the standard high level programming languages, namely ALGOL 60 (1), ALGOL 68 (2), FORTRAN (3), COBOL (4). In this case the algorithms must conform to the appropriate standard. If algorithms are submitted in other programming languages, the reference document for that language must be stated.

Algorithms must be self-contained. This means that an algorithm must consist of one or more complete segments, and that an algorithm must not use any non-local identifiers other than standard function names. COMMON areas are permitted in FORTRAN, but their use must be clearly described.

The algorithm must be written for publication in the appropriate reference language, and preceded by an appropriate Author's Note. It must be submitted in duplicate and be typewritten double-spaced. Where material is to appear in bold face it should be underlined in black. Where the appropriate character does not exist on a typewriter, it should be inserted neatly by hand in black and not be replaced by a similar composite character (e.g. $\leqslant$ should not be inserted as $\leq$).

An algorithm must be accompanied by a computer printout of a driver program testing it (possibly against test data) and producing test results. The machine, compiler and operating system used should be indicated. A computer readable copy of the algorithm, the test driver and any test data will be requested later, but should not be sent in the first instance. The Author's Note should include the theory of the method, with references, and also explain any tests used to verify the algorithm.

The algorithm must be syntactically correct, produce the results claimed and use computer resources as efficiently as possible. Constructions whose results may depend on the compiler used should be avoided (e.g. $y := x + f(x)$ where $f(b)$ is a function which alters the value of b). Comments should be used wherever appropriate to clarify the logic. Cases of failure should be clearly anticipated and handled. Approximate numerical constants must be given with as much accuracy as is appropriate. Numerical labels should be in ascending order.

Every effort is made to see that published algorithms are completely reliable. In particular all algorithms are submitted to independent referees and extensively checked. However, Notes or Correspondence which point out defects in or suggest improvements to previously published algorithms are welcomed. To help in preventing printing mistakes, galley proofs will be sent to authors where possible.

Whilst every effort is made to publish correct algorithms, no liability is assumed by any contributor, the Editor or *The Computer Journal* in connection therewith.

The copyright of all published algorithms remains with *The Computer Journal*. Nevertheless the reproduction of algorithms is explicitly permitted without charge provided that where the algorithm is used in another publication, reference is made to the author and to *The Computer Journal*.

In the event of the formation of a National Algorithm Library, all algorithms which have appeared in *The Computer Journal* will be made available to this Library.

References

1. PROGRAMMING LANGUAGE ALGOL, ISO/R/1538.
2. REVISED REPORT ON THE ALGORITHMIC LANGUAGE ALGOL 68. (1976). Edited by A. van Wijngaarden *et al*, Springer-Verlag.
3. PROGRAMMING LANGUAGE FORTRAN, ISO/R/1539.
4. PROGRAMMING LANGUAGE COBOL, ISO/R/1989.

Documents 1, 3 and 4 above may be obtained from: British Standards Institution, Sales Branch, 101 Pentonville Road, London N1.

Editors: P. A. Samet and A. C. Day, Computer Centre, University College London, 19 Gordon Street, London WC1H 0AH. Tel: 01-387 0858.