A Quantile Finding Program

by Luis Canales

In a series of papers included in this report, Claerbout has shown that the use of quantiles and medians could produce better (robust) results in source waveform estimation, deconvolution, entropy calculations, etc. In view of its importance it seems then appropriate to include a very efficient program for quantile and median finding. The program is a FORTRAN translation of a slightly modified ALGOL program due to Hoare (1971). The execution time grows linearly with the array's size, and the routine is extremely fast ; except for some short arrays, where some of the very efficient SORT routines could prove to be even faster.

Reference:

Hoare, C.A.R. (1971), Proof of a program find, Comm. of the A.C.M., Vol. 14, pp. 39-45 (January).

```
      $WATORV
            REAL A(10)/3.,1.,8.,9.,0.,2.,4.,7.,5.,6./
            DO 10 K=1,10
            CALL QUANT(K,A,10)
         10 PRINT,'A(',K,') =',A(K)
            STOP
            END
            SUBROUTINE QUANT( K , A , N )
C******
            REAL A(N)
            INTEGER LOW , HI
C******
C    THIS ROUTINE REORDERS THE ARRAY  ' A(N) ' ,SUCH THAT:
C          IF  I .LT. K   THEN   A(I) .LT. A(K)
C    IN OTHER WORDS A(K) IS THE (K/N)TH QUANTILE.
C       IF YOU WANT SEVERAL QUANTILES , START WITH THE
C    LARGEST , THEN CALL 'QUANT' WITH THE REDUCED ARRAY.
C    EVERY NEW CALL WILL BE FASTER.  TIME INCREASES LINEARLY.
C**********************************************************
            LOW = 1
            HI = N
          1 IF( LOW .GE. HI ) RETURN
            AK = A(K)
            I = LOW
            J = HI
          2 IF( A(I) .GE. AK ) GO TO 3
            I = I + 1
            GO TO 2
          3 IF( A(J) .LE. AK ) GO TO 4
            J = J - 1
            GO TO 3
          4 IF( I .GT. J ) GO TO 5
            AA = A(I)
            A(I) = A(J)
            A(J) = AA
            I = I + 1
            J = J -1
            IF( I .LE. J ) GO TO 2
          5 IF( J .LT. K ) LOW = I
            IF( K .LT. I ) HI = J
            GO TO 1
            END
      $DATA
      STOP
? cgo
COMPILE TIME =      0.01 SECONDS, OBJECT CODE=     1,336 BYTES, ARRAY AREA=
A(             1 ) =     0.0000000E 00
A(             2 ) =     0.1000000E 01
A(             3 ) =     0.2000000E 01
A(             4 ) =     0.3000000E 01
A(             5 ) =     0.4000000E 01
A(             6 ) =     0.5000000E 01
A(             7 ) =     0.6000000E 01
A(             8 ) =     0.7000000E 01
A(             9 ) =     0.8000000E 01
A(            10 ) =     0.9000000E 01
STOP IN LINE      6.
EXECUTION TIME =      0.03 SECONDS
?
```