M01DAF – NAG Fortran Library Routine Document

Note. Before using this routine, please read the Users' Note for your implementation to check the interpretation of bold italicised terms and other implementation-dependent details.

1 Purpose

M01DAF ranks a vector of ***real*** numbers in ascending or descending order.

2 Specification

```
SUBROUTINE M01DAF(RV, M1, M2, ORDER, IRANK, IFAIL)
INTEGER           M1, M2, IRANK(M2), IFAIL
real              RV(M2)
CHARACTER*1       ORDER
```

3 Description

M01DAF uses a variant of list-merging, as described by Knuth [1] pp 165-166. The routine takes advantage of natural ordering in the data, and uses a simple list insertion in a preparatory pass to generate ordered lists of length at least 10. The ranking is stable: equal elements preserve their ordering in the input data.

4 References

[1] Knuth D E (1973) *The Art of Computer Programming (Volume 3)* Addison–Wesley (2nd Edition)

5 Parameters

1: RV(M2) — ***real*** array *Input*

On entry: elements M1 to M2 of RV must contain ***real*** values to be ranked.

2: M1 — INTEGER *Input*

On entry: the index of the first element of RV to be ranked.

Constraint: M1 > 0.

3: M2 — INTEGER *Input*

On entry: the index of the last element of RV to be ranked.

Constraint: M2 $\geq$ M1.

4: ORDER — CHARACTER*1 *Input*

On entry: if ORDER is 'A', the values will be ranked in ascending (i.e., non-decreasing) order; if ORDER is 'D', into descending order.

Constraint: ORDER = 'A' or 'D'.

5: IRANK(M2) — INTEGER array *Output*

On exit: elements M1 to M2 of IRANK contain the ranks of the corresponding elements of RV. Note that the ranks are in the range M1 to M2: thus, if RV(i) is the first element in the rank order, IRANK(i) is set to M1.

6: IFAIL — INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.

On exit: IFAIL = 0 unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL = 0 or −1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors detected by the routine:

IFAIL = 1

On entry, M2 < 1,
or M1 < 1,
or M1 > M2.

IFAIL = 2

On entry, ORDER is not 'A' or 'D'.

7 Accuracy

Not applicable.

8 Further Comments

The average time taken by the routine is approximately proportional to $n \times \log n$, where $n = \text{M2} - \text{M1} + 1$.

9 Example

The example program reads a list of ***real*** numbers and ranks them in ascending order.

9.1 Program Text

Note. The listing of the example program presented below uses bold italicised terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     M01DAF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NMAX
      PARAMETER        (NMAX=100)
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, N
*     .. Local Arrays ..
      real             RV(NMAX)
      INTEGER          IRANK(NMAX)
*     .. External Subroutines ..
      EXTERNAL         M01DAF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'M01DAF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.GE.1 .AND. N.LE.NMAX) THEN
         READ (NIN,*) (RV(I),I=1,N)
         IFAIL = 0
*
         CALL M01DAF(RV,1,N,'Ascending',IRANK,IFAIL)
```

```
*
         WRITE (NOUT,*)
         WRITE (NOUT,*) '   Data   Ranks'
         WRITE (NOUT,*)
         DO 20 I = 1, N
            WRITE (NOUT,99999) RV(I), IRANK(I)
   20    CONTINUE
      END IF
      STOP
*
99999 FORMAT (1X,F7.1,I7)
      END
```

9.2 Program Data

```
M01DAF Example Program Data
12
5.3 4.6 7.8 1.7 5.3 9.9 3.2 4.3 7.8 4.5 1.2 7.6
```

9.3 Program Results

```
 M01DAF Example Program Results

    Data   Ranks

     5.3      7
     4.6      6
     7.8     10
     1.7      2
     5.3      8
     9.9     12
     3.2      3
     4.3      4
     7.8     11
     4.5      5
     1.2      1
     7.6      9
```