NAG Fortran Library Routine Document

M01DEF

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

M01DEF ranks the rows of a matrix of ***real*** numbers in ascending or descending order.

2 Specification

```
SUBROUTINE M01DEF(RM, LDM, M1, M2, N1, N2, ORDER, IRANK, IFAIL)
INTEGER           LDM, M1, M2, N1, N2, IRANK(M2), IFAIL
real              RM(LDM,N2)
CHARACTER*1       ORDER
```

3 Description

M01DEF ranks rows M1 to M2 of a matrix, using the data in columns N1 to N2 of those rows. The ordering is determined by first ranking the data in column N1, then ranking any tied rows according to the data in column $\mathrm{N1}+1$, and so on up to column N2.

M01DEF uses a variant of list-merging, as described by Knuth (1973), 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 rows preserve their ordering in the input data.

4 References

Knuth D E (1973) *The Art of Computer Programming (Volume 3)* (2nd Edition) Addison-Wesley

5 Parameters

1: RM(LDM,N2) – ***real*** array *Input*

On entry: columns N1 to N2 of rows M1 to M2 of RM must contain ***real*** data to be ranked.

2: LDM – INTEGER *Input*

On entry: the first dimension of the array RM as declared in the (sub)program from which M01DEF is called.

Constraint: $\mathrm{LDM} \geq \mathrm{M2}$.

3: M1 – INTEGER *Input*

On entry: the index of the first row of RM to be ranked.

Constraint: $\mathrm{M1} > 0$.

4: M2 – INTEGER *Input*

On entry: the index of the last row of RM to be ranked.

Constraint: $\mathrm{M2} \geq \mathrm{M1}$.

5: N1 – INTEGER *Input*

On entry: the index of the first column of RM to be used.

Constraint: $\mathrm{N1} > 0$.

6: N2 – INTEGER *Input*

On entry: the index of the last column of RM to be used.

Constraint: $\mathrm{N2} \geq \mathrm{N1}$.

7: ORDER – CHARACTER*1 *Input*

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

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

8: IRANK(M2) – INTEGER array *Output*

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

9: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

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

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, for users not familiar with this parameter the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

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 or warnings detected by the routine:

IFAIL = 1

On entry, $\mathrm{M2} < 1$,
or $\mathrm{N2} < 1$,
or $\mathrm{M1} < 1$,
or $\mathrm{M1} > \mathrm{M2}$,
or $\mathrm{N1} < 1$,
or $\mathrm{N1} > \mathrm{N2}$,
or $\mathrm{LDM} < \mathrm{M2}$.

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 $n = \mathrm{M2} - \mathrm{M1} + 1$.

9 Example

The example program reads a matrix of ***real*** numbers and ranks the rows 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.

```
*     M01DEF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          MMAX, NMAX
      PARAMETER        (MMAX=20,NMAX=20)
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, J, M, N
*     .. Local Arrays ..
      real             RM(MMAX,NMAX)
      INTEGER          IRANK(MMAX)
*     .. External Subroutines ..
      EXTERNAL         M01DEF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'M01DEF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) M, N
      IF (M.GE.1 .AND. M.LE.MMAX .AND. N.GE.1 .AND. N.LE.NMAX) THEN
         DO 20 I = 1, M
            READ (NIN,*) (RM(I,J),J=1,N)
   20    CONTINUE
         IFAIL = 0
*
         CALL M01DEF(RM,MMAX,1,M,1,N,'Ascending',IRANK,IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Data                        Ranks'
         WRITE (NOUT,*)
         DO 40 I = 1, M
            WRITE (NOUT,99999) (RM(I,J),J=1,N), IRANK(I)
   40    CONTINUE
      END IF
      STOP
*
99999 FORMAT (1X,3F7.1,I11)
      END
```

9.2 Program Data

```
M01DEF Example Program Data
12 3
6.0 5.0 4.0
5.0 2.0 1.0
2.0 4.0 9.0
4.0 9.0 6.0
4.0 9.0 5.0
4.0 1.0 2.0
3.0 4.0 1.0
2.0 4.0 6.0
1.0 6.0 4.0
9.0 3.0 2.0
6.0 2.0 5.0
4.0 9.0 6.0
```

9.3 Program Results

```
 M01DEF Example Program Results

 Data                        Ranks

     6.0    5.0    4.0         11
     5.0    2.0    1.0          9
     2.0    4.0    9.0          3
     4.0    9.0    6.0          7
     4.0    9.0    5.0          6
     4.0    1.0    2.0          5
     3.0    4.0    1.0          4
     2.0    4.0    6.0          2
     1.0    6.0    4.0          1
     9.0    3.0    2.0         12
     6.0    2.0    5.0         10
     4.0    9.0    6.0          8
```