

# NAG Fortran Library Routine Document

## M01DJF

**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

M01DJF ranks the columns of a matrix of *real* numbers in ascending or descending order.

### 2 Specification

```
SUBROUTINE M01DJF(RM, LDM, M1, M2, N1, N2, ORDER, IRANK, IFAIL)
INTEGER          LDM, M1, M2, N1, N2, IRANK(N2), IFAIL
real           RM(LDM,N2)
CHARACTER*1     ORDER
```

### 3 Description

M01DJF ranks columns N1 to N2 of a matrix, using the data in rows M1 to M2 of those columns. The ordering is determined by first ranking the data in row M1, then ranking any tied columns according to the data in row M1 + 1, and so on up to row M2.

M01DJF 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 columns 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:* rows M1 to M2 of columns N1 to N2 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 M01DJF is called.  
*Constraint:* LDM ≥ M2.
- 3: M1 – INTEGER *Input*  
*On entry:* the index of the first row of RM to be used.  
*Constraint:* M1 > 0.
- 4: M2 – INTEGER *Input*  
*On entry:* the index of the last row of RM to be used.  
*Constraint:* M2 ≥ M1.

- 5: N1 – INTEGER *Input*  
*On entry:* the index of the first column of RM to be ranked.  
*Constraint:*  $N1 > 0$ .
- 6: N2 – INTEGER *Input*  
*On entry:* the index of the last column of RM to be ranked.  
*Constraint:*  $N2 \geq N1$ .
- 7: ORDER – CHARACTER\*1 *Input*  
*On entry:* if ORDER is 'A', the columns will be ranked in ascending (i.e., non-decreasing) order; if ORDER is 'D', into descending order.  
*Constraint:* ORDER = 'A' or 'D'.
- 8: IRANK(N2) – INTEGER array *Output*  
*On exit:* elements N1 to N2 of IRANK contain the ranks of the corresponding columns of RM. Note that the ranks are in the range N1 to N2: thus, if the  $i$ th column of RM is the first in the rank order, IRANK( $i$ ) is set to N1.
- 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, M2 < 1,  
 or N2 < 1,  
 or M1 < 1,  
 or M1 > M2,  
 or N1 < 1,  
 or N1 > N2,  
 or LDM < 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  $n = N2 - N1 + 1$ .

## 9 Example

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

```
*      M01DJF 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(NMAX)
*      .. External Subroutines ..
      EXTERNAL        M01DJF
*      .. Executable Statements ..
      WRITE (NOUT,*) 'M01DJF 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 M01DJF(RM,MMAX,1,M,1,N,'Ascending',IRANK,IFAIL)
*
          WRITE (NOUT,*)
          WRITE (NOUT,*) 'Data'
          WRITE (NOUT,*)
          DO 40 I = 1, M
            WRITE (NOUT,99999) (RM(I,J),J=1,N)
40      CONTINUE
          WRITE (NOUT,*)
          WRITE (NOUT,*) 'Ranks'
          WRITE (NOUT,*)
          WRITE (NOUT,99998) (IRANK(I),I=1,N)
        END IF
      STOP
*
99999 FORMAT (1X,12F6.1)
99998 FORMAT (1X,12I6)
      END
```

### 9.2 Program Data

```
M01DJF Example Program Data
3 12
5.0 4.0 3.0 2.0 2.0 1.0 9.0 4.0 4.0 2.0 2.0 1.0
3.0 8.0 2.0 5.0 5.0 6.0 9.0 8.0 9.0 5.0 4.0 1.0
9.0 1.0 6.0 1.0 2.0 4.0 8.0 1.0 2.0 2.0 6.0 2.0
```

### 9.3 Program Results

M01DJF Example Program Results

Data

5.0	4.0	3.0	2.0	2.0	1.0	9.0	4.0	4.0	2.0	2.0	1.0
3.0	8.0	2.0	5.0	5.0	6.0	9.0	8.0	9.0	5.0	4.0	1.0
9.0	1.0	6.0	1.0	2.0	4.0	8.0	1.0	2.0	2.0	6.0	2.0

Ranks

11	8	7	4	5	2	12	9	10	6	3	1
----	---	---	---	---	---	----	---	----	---	---	---

---