M01ZAF – 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

M01ZAF inverts a permutation, and hence converts a rank vector to an index vector, or vice versa.

2 Specification

```
SUBROUTINE M01ZAF(IPERM, M1, M2, IFAIL)
INTEGER           IPERM(M2), M1, M2, IFAIL
```

3 Description

There are two common ways of describing a permutation using an integer vector IPERM. The first uses ranks: IPERM(i) holds the position to which the ith data element should be moved in order to sort the data; in other words its rank in the sorted order. The second uses indices: IPERM(i) holds the current position of the data element which would occur in ith position in sorted order. For example, given the values

3.5	5.9	2.9	0.5

to be sorted in ascending order, the ranks would be

3	4	2	1

and the indices would be

4	3	1	2

The M01D- routines generate ranks, and the M01E- routines require ranks to be supplied to specify the re-ordering. However if it is desired simply to refer to the data in sorted order without actually re-ordering them, indices are more convenient than ranks (see the example in Section 9).

M01ZAF can be used to convert ranks to indices, or indices to ranks, as the two permutations are inverses of one another.

4 References

None.

5 Parameters

1: IPERM(M2) — INTEGER array *Input/Output*

On entry: elements M1 to M2 of IPERM must contain a permutation of the integers M1 to M2.

On exit: these elements contain the inverse permutation of the integers M1 to M2.

2: M1 — INTEGER *Input*

3: M2 — INTEGER *Input*

On entry: M1 and M2 must specify the range of elements used in the array IPERM and the range of values in the permutation, as specified under IPERM.

Constraint: $0 < \mathrm{M1} \leq \mathrm{M2}$.

4: 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

Elements M1 to M2 of IPERM contain a value outside the range M1 to M2.

IFAIL = 3

Elements M1 to M2 of IPERM contain a repeated value.

If IFAIL = 2 or 3, elements M1 to M2 of IPERM do not contain a permutation of the integers M1 to M2; on exit these elements are usually corrupted. To check the validity of a permutation without the risk of corrupting it, use M01ZBF.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

The example program reads a matrix of ***real*** numbers and prints its rows in ascending order as ranked by M01DEF. The program first calls M01DEF to rank the rows, and then calls M01ZAF to convert the rank vector to an index vector, which is used to refer to the rows in sorted 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.

```
*     M01ZAF 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           IPERM(MMAX)
*     .. External Subroutines ..
      EXTERNAL          M01DEF, M01ZAF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'M01ZAF 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',IPERM,IFAIL)
         CALL M01ZAF(IPERM,1,M,IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Matrix sorted by rows'
         WRITE (NOUT,*)
         DO 40 I = 1, M
            WRITE (NOUT,99999) (RM(IPERM(I),J),J=1,N)
   40    CONTINUE
      END IF
      STOP
*
99999 FORMAT (1X,3F7.1)
      END
```

9.2 Program Data

```
M01ZAF 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

```
 M01ZAF Example Program Results

 Matrix sorted by rows

     1.0    6.0    4.0
     2.0    4.0    6.0
     2.0    4.0    9.0
```

```
      3.0    4.0    1.0
      4.0    1.0    2.0
      4.0    9.0    5.0
      4.0    9.0    6.0
      4.0    9.0    6.0
      5.0    2.0    1.0
      6.0    2.0    5.0
      6.0    5.0    4.0
      9.0    3.0    2.0
```