NAG Fortran Library Routine Document

F01ZBF

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

F01ZBF copies a complex triangular matrix stored in a packed one-dimensional array into an unpacked two-dimensional array, or vice versa.

2 Specification

```
SUBROUTINE F01ZBF(JOB, UPLO, DIAG, N, A, LDA, B, IFAIL)
INTEGER           N, LDA, IFAIL
complex           A(LDA,N), B((N*(N+1))/2)
CHARACTER*1       JOB, UPLO, DIAG
```

3 Description

F01ZBF unpacks a triangular matrix stored in a vector into a two-dimensional array, or packs a triangular matrix stored in a two-dimensional array into a vector. The matrix is packed by column. This routine is intended for possible use in conjunction with routines from Chapters F06, F07 and F08, where some routines that use triangular matrices store them in the packed form described below.

4 References

None.

5 Parameters

1: JOB – CHARACTER*1 *Input*

On entry: specifies whether the triangular matrix is to be packed or unpacked, as follows:

if JOB = 'P' (Pack), the matrix is to be packed into array B;

if JOB = 'U' (Unpack), the matrix is to be unpacked into array A.

Constraint: JOB must be one of 'P' or 'U'.

2: UPLO – CHARACTER*1 *Input*

On entry: the type of the matrix to be copied, as follows:

if UPLO = 'L' (Lower), the matrix is lower triangular. In this case the packed vector holds, or will hold on exit, the matrix elements in the following order: $(1,1),(2,1),\ldots,(N,1),(2,2),(3,2),\ldots,(N,2)$, etc.;

if UPLO='U' or 'u' (Upper), the matrix is upper triangular. In this case the packed vector holds, or will hold on exit, the matrix elements in the following order: (1,1), (1,2), (2,2), (1,3), (2,3), (3,3), (1,4), etc.

Constraint: UPLO must be one of 'L' or 'U'.

3: DIAG – CHARACTER*1 *Input*

On entry: DIAG must specify whether the diagonal elements of the matrix are to be copied, as follows:

if DIAG = 'B' (Blank), the diagonal elements of the matrix are not referenced and not copied;

if DIAG = 'U' (Unit diagonal), the diagonal elements of the matrix are not referenced, but are assumed all to be unity, and are copied as such.

if DIAG = 'N' (Non-unit diagonal), the diagonal elements of the matrix are referenced and copied.

Constraint: DIAG must be one of 'B', 'U' or 'N'.

4: N – INTEGER *Input*

On entry: N must specify the number of rows and columns of the triangular matrix.

Constraint: $N > 0$.

5: A(LDA,N) – ***complex*** array *Input/Output*

On entry: if JOB = 'P', then the leading N by N part of A must contain the matrix to be copied, stored in unpacked form, in the upper or lower triangle depending on parameter UPLO. The opposite triangle of A is not referenced and need not be assigned.

On exit: if JOB = 'U', then the leading N by N part of array A contains the copied matrix, stored in unpacked form, in the upper or lower triangle depending on parameter UPLO. The opposite triangle of A is not referenced.

6: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F01ZBF is called.

Constraint: $LDA \geq N$.

7: B((N*(N+1))/2) – ***complex*** array *Input/Output*

On entry: if JOB = 'U', then B must contain the triangular matrix packed by column.

On exit: if JOB = 'P', then B contains the triangular matrix packed by column.

Note that B must have space for the diagonal elements of the matrix, even if these are not stored.

8: 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, JOB $\neq$ 'P' or 'U'.

IFAIL $= 2$

On entry, UPLO $\neq$ 'L' or 'U'.

IFAIL $= 3$

On entry, DIAG $\neq$ 'N', 'U' or 'B'.

IFAIL $= 4$

On entry, N < 1.

IFAIL $= 5$

On entry, LDA $<$ N.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

This example program reads in a triangular matrix A, and copies it to the packed matrix B.

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.

```
*     F01ZBF Example Program Text
*     Mark 14 Release.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, LDA, LENB
      PARAMETER        (NMAX=10,LDA=NMAX,LENB=(NMAX*(NMAX+1))/2)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, J, LB, N
      CHARACTER        DIAG, UPLO
*     .. Local Arrays ..
      complex          A(LDA,NMAX), B(LENB)
      CHARACTER        CLABS(1), RLABS(1)
*     .. External Subroutines ..
      EXTERNAL         F01ZBF, X04DBF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F01ZBF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      WRITE (NOUT,*)
      READ (NIN,*) N, UPLO, DIAG
*     Read a triangular matrix of order N
      DO 20 I = 1, N
         READ (NIN,*) (A(I,J),J=1,N)
   20 CONTINUE
      IFAIL = 0
*     Print the unpacked matrix
      CALL X04DBF(UPLO,DIAG,N,N,A,LDA,'B','F5.2','Unpacked Matrix A:',
     +            'I',RLABS,'I',CLABS,80,0,IFAIL)
      WRITE (NOUT,*)
*
*     Convert to packed vector form
      CALL F01ZBF('Pack',UPLO,DIAG,N,A,LDA,B,IFAIL)
```

```
*
      LB = N*(N+1)/2
*     Print the packed vector
      CALL X04DBF('G','X',LB,1,B,LB,'B','F5.2','Packed Vector B:','I',
     +            RLABS,'N',CLABS,80,0,IFAIL)
      STOP
      END
```

9.2 Program Data

```
F01ZBF Example Program Data.
4 'U' 'N'                                          N  UPLO  DIAG
 (1.1,1.1)  (1.2,1.2)  (1.3,1.3)  (1.4,1.4)     Unpacked Matrix A
 (0.0,0.0)  (2.2,2.2)  (2.3,2.3)  (2.4,2.4)
 (0.0,0.0)  (0.0,0.0)  (3.3,3.3)  (3.4,3.4)
 (0.0,0.0)  (0.0,0.0)  (0.0,0.0)  (4.4,4.4)
```

9.3 Program Results

```
 F01ZBF Example Program Results

 Unpacked Matrix A:
                1             2             3             4
 1  ( 1.10, 1.10) ( 1.20, 1.20) ( 1.30, 1.30) ( 1.40, 1.40)
 2                ( 2.20, 2.20) ( 2.30, 2.30) ( 2.40, 2.40)
 3                              ( 3.30, 3.30) ( 3.40, 3.40)
 4                                            ( 4.40, 4.40)

 Packed Vector B:
  1  ( 1.10, 1.10)
  2  ( 1.20, 1.20)
  3  ( 2.20, 2.20)
  4  ( 1.30, 1.30)
  5  ( 2.30, 2.30)
  6  ( 3.30, 3.30)
  7  ( 1.40, 1.40)
  8  ( 2.40, 2.40)
  9  ( 3.40, 3.40)
 10  ( 4.40, 4.40)
```