

# NAG Library Routine Document

## F01VFF (ZTRTTF)

**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

F01VFF (ZTRTTF) copies a complex triangular matrix stored full format in a two-dimensional array to Rectangular Full Packed (RFP) format. The RFP storage format is described in Section 3.3.3 in the F07 Chapter Introduction.

### 2 Specification

```
SUBROUTINE F01VFF (TRANSR, UPLO, N, A, LDA, ARF, INFO)
INTEGER          N, LDA, INFO
COMPLEX (KIND=nag_wp) A(LDA,*), ARF(N*(N+1)/2)
CHARACTER(1)     TRANSR, UPLO
```

The routine may be called by its LAPACK name *ztrtf*.

### 3 Description

F01VFF (ZTRTTF) packs a complex  $n$  by  $n$  triangular matrix  $A$ , stored conventionally in a two-dimensional array into RFP format. This routine is intended for possible use in conjunction with routines from Chapters F06 and F07 where some routines that use triangular matrices store them in RFP format.

### 4 References

None.

### 5 Parameters

- 1: TRANSR – CHARACTER(1) *Input*  
*On entry:* specifies whether the normal RFP representation of  $A$  or its conjugate transpose is stored.  
 TRANSR = 'N'  
 The matrix  $A$  is stored in normal RFP format.  
 TRANSR = 'C'  
 The conjugate transpose of the RFP representation of the matrix  $A$  is stored.  
*Constraint:* TRANSR = 'N' or 'C'.
- 2: UPLO – CHARACTER(1) *Input*  
*On entry:* specifies whether  $A$  is upper or lower triangular.  
 UPLO = 'U'  
 $A$  is upper triangular.  
 UPLO = 'L'  
 $A$  is lower triangular.  
*Constraint:* UPLO = 'U' or 'L'.

- 3: N – INTEGER *Input*  
*On entry:*  $n$ , the order of the matrix  $A$ .  
*Constraint:*  $N \geq 0$ .
- 4: A(LDA,\*) – COMPLEX (KIND=nag\_wp) array *Input*  
**Note:** the second dimension of the array  $A$  must be at least  $N$ .  
*On entry:* the triangular matrix  $A$ .  
 If UPLO = 'U',  $A$  is upper triangular and the elements of the array below the diagonal are not referenced.  
 If UPLO = 'L',  $A$  is lower triangular and the elements of the array above the diagonal are not referenced.
- 5: LDA – INTEGER *Input*  
*On entry:* the first dimension of the array  $A$  as declared in the (sub)program from which F01VFF (ZTRTTF) is called.  
*Constraint:*  $LDA \geq \max(1, N)$ .
- 6: ARF( $N \times (N + 1)/2$ ) – COMPLEX (KIND=nag\_wp) array *Output*  
*On exit:* the triangular matrix  $A$  in RFP format, as described in Section 3.3.3 in the F07 Chapter Introduction.
- 7: INFO – INTEGER *Output*  
*On exit:* INFO = 0 unless the routine detects an error (see Section 6).

## 6 Error Indicators and Warnings

Errors or warnings detected by the routine:

INFO < 0

If INFO =  $-i$ , argument  $i$  had an illegal value. An explanatory message is output, and execution of the program is terminated.

## 7 Accuracy

Not applicable.

## 8 Further Comments

None.

## 9 Example

This example reads in a triangular matrix and copies it to RFP format.

### 9.1 Program Text

```

Program f01vffe
!      F01VFF Example Program Text
!
!      Mark 24 Release. NAG Copyright 2012.
!
!      .. Use Statements ..
!      Use nag_library, Only: nag_wp, x04dbf, ztrttf

```

```

! .. Implicit None Statement ..
Implicit None
! .. Parameters ..
Integer, Parameter      :: incl = 1, indent = 0, ncols = 80,    &
                        nin = 5, nout = 6
Character (1), Parameter  :: brac = 'B', diag = 'N', intlabel =  &
                        'I', matrix = 'G', nolabel = 'N'
Character (4), Parameter  :: form = 'F5.2'
! .. Local Scalars ..
Integer                :: i, ifail, info, lda, lenarf, n
Character (21)         :: title
Character (1)          :: transr, uplo
! .. Local Arrays ..
Complex (Kind=nag_wp), Allocatable :: a(:,,:), arf(:)
Character (1)          :: clabs(1), rlabs(1)
! .. Executable Statements ..
Write (nout,*) 'F01VFF Example Program Results'
Skip heading in data file
Read (nin,*)
Write (nout,*)
Flush (nout)
Read (nin,*) n, uplo, transr
lda = n
lenarf = n*(n+1)/2
Allocate (a(lda,n),arf(lenarf))

! Read a triangular matrix of order n into array A
Do i = 1, n
  Read (nin,*) a(i,1:n)
End Do

! Print the unpacked array
title = 'Unpacked Matrix A:  '
ifail = 0
Call x04dbf(uplo,diag,n,n,a,lda,brac,form,title,intlabel,rlabs,intlabel, &
  clabs,ncols,indent,ifail)

Write (nout,*)
Flush (nout)

! Convert to Rectangular Full Packed form
info = 0
! The NAG name equivalent of ztrttf is f01vff
Call ztrttf(transr,uplo,n,a,lda,arf,info)

! Print the Rectangular Full Packed array
title = 'RFP Packed Array ARF:'
ifail = 0
Call x04dbf(matrix,diag,lenarf,incl,arf,lenarf,brac,form,title,intlabel, &
  rlabs,nolabel,clabs,ncols,indent,ifail)

End Program f01vffe

```

## 9.2 Program Data

F01VFF Example Program Data

```

4 'U' 'N'                                : n, uplo, transr
(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

F01VFF 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)

RFP Packed Array ARF:

1 ( 1.30, 1.30)  
2 ( 2.30, 2.30)  
3 ( 3.30, 3.30)  
4 ( 1.10, -1.10)  
5 ( 1.20, -1.20)  
6 ( 1.40, 1.40)  
7 ( 2.40, 2.40)  
8 ( 3.40, 3.40)  
9 ( 4.40, 4.40)  
10 ( 2.20, -2.20)

---