NAG Library Routine Document

F06TCF

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

F06TCF performs the matrix-vector operation

$$y \leftarrow \alpha Ax + \beta y$$

where A is an n by n complex symmetric matrix stored in packed form, x and y are n-element complex vectors, and α and β are complex scalars.

2 Specification

```
SUBROUTINE F06TCF (UPLO, N, ALPHA, AP, X, INCX, BETA, Y, INCY)
INTEGER                 N, INCX, INCY
COMPLEX (KIND=nag_wp) ALPHA, AP(*), X(*), BETA, Y(*)
CHARACTER(1)            UPLO
```

3 Description

None.

4 References

None.

5 Parameters

1: UPLO – CHARACTER(1) *Input*

On entry: specifies whether the upper or lower triangular part of A is stored.

UPLO = 'U'
The upper triangular part of A is stored.

UPLO = 'L'
The lower triangular part of A is stored.

Constraint: UPLO = 'U' or 'L'.

2: N – INTEGER *Input*

On entry: n, the order of the matrix A.

Constraint: N ≥ 0.

3: ALPHA – COMPLEX (KIND=nag_wp) *Input*

On entry: the scalar α.

4: AP$(*)$ – COMPLEX (KIND=nag_wp) array *Input*

Note: the dimension of the array AP must be at least N $\times$ (N + 1)/2.

On entry: the n by n symmetric matrix A, packed by columns.

More precisely,

if UPLO = 'U', the upper triangle of A must be stored with element A_{ij} in AP$(i+j(j-1)/2)$ for $i \leq j$;

if UPLO = 'L', the lower triangle of A must be stored with element A_{ij} in AP$(i+(2n-j)(j-1)/2)$ for $i \geq j$.

5: X$(*)$ – COMPLEX (KIND=nag_wp) array *Input*

Note: the dimension of the array X must be at least $\max(1, 1+(\text{N}-1)\times|\text{INCX}|)$.

On entry: the n-element vector x.

If INCX > 0, x_i must be stored in X$(1+(i-1)\times\text{INCX})$, for $i = 1, 2, \ldots, \text{N}$.

If INCX < 0, x_i must be stored in X$(1-(\text{N}-i)\times\text{INCX})$, for $i = 1, 2, \ldots, \text{N}$.

Intermediate elements of X are not referenced.

6: INCX – INTEGER *Input*

On entry: the increment in the subscripts of X between successive elements of x.

Constraint: INCX $\neq 0$.

7: BETA – COMPLEX (KIND=nag_wp) *Input*

On entry: the scalar β.

8: Y$(*)$ – COMPLEX (KIND=nag_wp) array *Input/Output*

Note: the dimension of the array Y must be at least $\max(1, 1+(\text{N}-1)\times|\text{INCY}|)$.

On entry: the n-element vector y.

If INCY > 0, y_i must be stored in Y$(1+(i-1)\times\text{INCY})$, for $i = 1, 2, \ldots, \text{N}$.

If INCY < 0, y_i must be stored in Y$(1-(\text{N}-i)\times\text{INCY})$, for $i = 1, 2, \ldots, \text{N}$.

On exit: the updated vector y stored in the array elements used to supply the original vector y.

9: INCY – INTEGER *Input*

On entry: the increment in the subscripts of Y between successive elements of y.

Constraint: INCY $\neq 0$.

6 Error Indicators and Warnings

None.

7 Accuracy

Not applicable.

8 Parallelism and Performance

Not applicable.

9 Further Comments

None.

10 Example

None.