

# NAG Library Routine Document

## F06TMF

**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

F06TMF performs a Unitary similarity transformation (as a sequence of plane rotations) of a complex Hermitian matrix.

### 2 Specification

SUBROUTINE F06TMF (UPLO, PIVOT, DIRECT, N, K1, K2, C, S, A, LDA)

INTEGER N, K1, K2, LDA  
 REAL (KIND=nag\_wp) C(\*)  
 COMPLEX (KIND=nag\_wp) S(\*), A(LDA,\*)  
 CHARACTER(1) UPLO, PIVOT, DIRECT

### 3 Description

F06TMF performs the transformation

$$A \leftarrow PAP^H$$

where  $A$  is an  $n$  by  $n$  complex Hermitian matrix, and  $P$  is a complex unitary matrix defined as a sequence of plane rotations,  $P_k$ , applied in planes  $k_1$  to  $k_2$ .

The 2 by 2 plane rotation part of  $P_k$  is assumed to have the form

$$\begin{pmatrix} c_k & \bar{s}_k \\ -s_k & c_k \end{pmatrix}$$

with  $c_k$  real.

### 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: PIVOT – CHARACTER(1) *Input*  
*On entry:* specifies the plane rotated by  $P_k$ .  
 PIVOT = 'V' (variable pivot)  
 $P_k$  rotates the  $(k, k + 1)$  plane.

- PIVOT = 'T' (top pivot)  
 $P_k$  rotates the  $(k_1, k + 1)$  plane.
- PIVOT = 'B' (bottom pivot)  
 $P_k$  rotates the  $(k, k_2)$  plane.
- Constraint:* PIVOT = 'V', 'T' or 'B'.
- 3: DIRECT – CHARACTER(1) *Input*  
*On entry:* specifies the sequence direction.  
 DIRECT = 'F' (forward sequence)  
 $P = P_{k_2-1} \cdots P_{k_1+1} P_{k_1}$ .  
 DIRECT = 'B' (backward sequence)  
 $P = P_{k_1} P_{k_1+1} \cdots P_{k_2-1}$ .  
*Constraint:* DIRECT = 'F' or 'B'.
- 4: N – INTEGER *Input*  
*On entry:*  $n$ , the order of the matrix  $A$ .  
*Constraint:*  $N \geq 0$ .
- 5: K1 – INTEGER *Input*  
 6: K2 – INTEGER *Input*  
*On entry:* the values  $k_1$  and  $k_2$ .  
 If  $K1 < 1$  or  $K2 \leq K1$  or  $K2 > N$ , an immediate return is effected.
- 7: C(\*) – REAL (KIND=nag\_wp) array *Input*  
**Note:** the dimension of the array C must be at least  $K2 - K1$ .  
*On entry:*  $C(k)$  must hold  $c_k$ , the cosine of the rotation  $P_k$ , for  $k = k_1, \dots, k_2 - 1$ .
- 8: S(\*) – COMPLEX (KIND=nag\_wp) array *Input*  
**Note:** the dimension of the array S must be at least  $K2 - K1$ .  
*On entry:*  $S(k)$  must hold  $s_k$ , the sine of the rotation  $P_k$ , for  $k = k_1, \dots, k_2 - 1$ .
- 9: A(LDA,\*) – COMPLEX (KIND=nag\_wp) array *Input/Output*  
**Note:** the second dimension of the array A must be at least  $\max(1, N)$ .  
*On entry:* the  $n$  by  $n$  Hermitian matrix  $A$ .  
 If UPLO = 'U', the upper triangular part of  $A$  must be stored and the elements of the array below the diagonal are not referenced.  
 If UPLO = 'L', the lower triangular part of  $A$  must be stored and the elements of the array above the diagonal are not referenced.  
*On exit:* the transformed matrix  $A$ . The imaginary parts of the diagonal elements are set to zero.
- 10: LDA – INTEGER *Input*  
*On entry:* the first dimension of the array A as declared in the (sub)program from which F06TMF is called.  
*Constraint:*  $LDA \geq \max(1, N)$ .

## 6 Error Indicators and Warnings

None.

**7 Accuracy**

Not applicable.

**8 Further Comments**

None.

**9 Example**

None.

---