NAG Fortran 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 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.

2 Specification

```
SUBROUTINE F06TMF (UPLO, PIVOT, DIRECT, N, K1, K2, C, S, A, LDA)
INTEGER             N, K1, K2, LDA
double precision    C(*)
complex*16          S(*), A(LDA,*)
CHARACTER*1         UPLO, PIVOT, DIRECT
```

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 as follows:

if UPLO = 'U', the upper triangular part of A is stored;
if 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:

if PIVOT = 'V' (variable pivot), P_k rotates the $(k, k+1)$ plane;
if PIVOT = 'T' (top pivot), P_k rotates the $(k_1, k+1)$ plane;
if 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:

if DIRECT = 'F' (forward sequence), $P = P_{k_2-1} \cdots P_{k_1+1} P_{k_1}$;
if 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: $\mathrm{N} \geq 0$.

5: K1 – INTEGER *Input*
6: K2 – INTEGER *Input*

On entry: the values k_1 and k_2.

7: C($*$) – ***double precision*** array *Input*

On entry: $\mathrm{C}(k)$ must hold c_k, the cosine of the rotation P_k, for $k = k_1, \ldots, k_2 - 1$.

8: S($*$) – ***complex*16*** array *Input*

On entry: $\mathrm{S}(k)$ must hold s_k, the sine of the rotation P_k, for $k = k_1, \ldots, k_2 - 1$.

9: A(LDA,$*$) – ***complex*16*** array *Input/Output*

Note: the second dimension of the array A must be at least $\max(1, \mathrm{N})$.

On entry: the n by n Hermitian matrix A. If UPLO = 'U', the upper triangle of A must be stored and the elements of the array below the diagonal are not referenced; if UPLO = 'L', the lower triangle 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: $\mathrm{LDA} \geq \max(1, \mathrm{N})$.

6 Error Indicators and Warnings

None.