NAG Fortran Library Routine Document

F06QXF

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

F06QXF performs the transformation

$$A \leftarrow PA \quad \text{or} \quad A \leftarrow AP^T,$$

where A is an m by n real matrix and P is a real orthogonal 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

$$\begin{pmatrix} c_k & s_k \\ -s_k & c_k \end{pmatrix}.$$

2 Specification

```
SUBROUTINE F06QXF (SIDE, PIVOT, DIRECT, M, N, K1, K2, C, S, A, LDA)
INTEGER             M, N, K1, K2, LDA
double precision    C(*), S(*), A(LDA,*)
CHARACTER*1         SIDE, PIVOT, DIRECT
```

3 Description

None.

4 References

None.

5 Parameters

1: SIDE – CHARACTER*1 *Input*

On entry: specifies whether A is operated on from the left or the right, as follows:

if SIDE = 'L', A is pre-multiplied from the left;
if SIDE = 'R', A is post-multiplied from the right.

Constraint: SIDE = 'L' or 'R'.

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: M – INTEGER *Input*

On entry: m, the number of rows of the matrix A.

Constraint: $\mathrm{M} \geq 0$.

5: N – INTEGER *Input*

On entry: n, the number of columns of the matrix A.

Constraint: $\mathrm{N} \geq 0$.

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

On entry: the values k_1 and k_2.

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

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

9: S($*$) – ***double precision*** array *Input*

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

10: A(LDA,$*$) – ***double precision*** array *Input/Output*

Note: the second dimension of the array A must be at least max(1, N).

On entry: the m by n matrix A.

On exit: the transformed matrix A.

11: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F06QXF is called.

Constraint: LDA $\geq$ max(1, M).

6 Error Indicators and Warnings

None.