

# NAG Library Routine Document

## F06VXF

**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

F06VXF applies to a complex rectangular matrix a sequence of plane rotations having real cosines and complex sines.

### 2 Specification

```
SUBROUTINE F06VXF (SIDE, PIVOT, DIRECT, M, N, K1, K2, C, S, A, LDA)
INTEGER                M, N, K1, K2, LDA
REAL (KIND=nag_wp)    C(*), S(*)
COMPLEX (KIND=nag_wp) A(LDA,*)
CHARACTER(1)          SIDE, PIVOT, DIRECT
```

### 3 Description

F06VXF performs the transformation

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

where  $A$  is an  $m$  by  $n$  complex matrix and  $P$  is a real orthogonal matrix, defined as a sequence of real 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

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

with  $c_k$  and  $s_k$  real.

### 4 References

None.

### 5 Arguments

- 1: SIDE – CHARACTER(1) *Input*  
*On entry:* specifies whether  $A$  is operated on from the left or the right.  
 SIDE = 'L'  
      $A$  is pre-multiplied from the left.  
 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  $P_k$ .  
 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: M – INTEGER *Input*  
*On entry:*  $m$ , the number of rows of the matrix  $A$ .  
*Constraint:*  $M \geq 0$ .
- 5: N – INTEGER *Input*  
*On entry:*  $n$ , the number of columns of the matrix  $A$ .  
*Constraint:*  $N \geq 0$ .
- 6: K1 – INTEGER *Input*  
7: K2 – INTEGER *Input*  
*On entry:* the values  $k_1$  and  $k_2$ .  
If  $K1 < 1$  or  $K2 \leq K1$ , or SIDE = 'L' and  $K2 > M$ , or SIDE = 'R' and  $K2 > N$ , an immediate return is effected.
- 8: 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  $k = k_1, \dots, k_2 - 1$ .
- 9: S(\*) – REAL (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  $k = k_1, \dots, k_2 - 1$ .
- 10: A(LDA,\*) – COMPLEX (KIND=nag\_wp) array *Input/Output*  
**Note:** the second dimension of the array A must be at least 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 F06VXF is called.  
*Constraint:*  $LDA \geq \max(1, M)$ .

## **6 Error Indicators and Warnings**

None.

## **7 Accuracy**

Not applicable.

## **8 Parallelism and Performance**

F06VXF is not threaded in any implementation.

## **9 Further Comments**

None.

## **10 Example**

None.

---