

# NAG Library Routine Document

## F06TWF

**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

F06TWF transforms a complex upper triangular matrix to an upper spiked matrix by applying a given sequence of plane rotations.

### 2 Specification

```
SUBROUTINE F06TWF (SIDE, 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)    SIDE
```

### 3 Description

F06TWF transforms an  $n$  by  $n$  complex upper triangular matrix  $U$  with real diagonal elements, to an upper spiked matrix  $H$ , by applying a given sequence of plane rotations from either the left or the right, in planes  $k_1$  to  $k_2$ .  $H$  has real diagonal elements except where the spike joins the diagonal.

If  $SIDE = 'L'$ ,  $H$  has a row spike, with nonzero elements  $h_{k_2,k}$ , for  $k = k_1, k_1 + 1, \dots, k_2 - 1$ . The rotations are applied from the left:

$$H = PU,$$

where  $P = P_{k_1} P_{k_1+1} \cdots P_{k_2-1}$  and  $P_k$  is a rotation in the  $(k, k_2)$  plane.

If  $SIDE = 'R'$ ,  $H$  has a column spike, with nonzero elements  $h_{k+1,k_1}$ , for  $k = k_1, k_1 + 1, \dots, k_2 - 1$ . The rotations are applied from the right:

$$HP^H = R,$$

where  $P = P_{k_2-1} \cdots P_{k_1+1} P_{k_1}$  and  $P_k$  is a rotation in the  $(k_1, k + 1)$  plane.

The 2 by 2 plane rotation part of  $P_k$  has the form

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

with  $c_k$  real.

### 4 References

None.

### 5 Arguments

- 1:  $SIDE$  – CHARACTER(1) *Input*  
*On entry:* specifies whether  $U$  is operated on from the left or the right.  
 $SIDE = 'L'$   
 $U$  is pre-multiplied from the left.

- SIDE = 'R'  
*U* is post-multiplied from the right.  
 Constraint: SIDE = 'L' or 'R'.
- 2: N – INTEGER *Input*  
 On entry:  $n$ , the order of the matrices *U* and *H*.  
 Constraint:  $N \geq 0$ .
- 3: K1 – INTEGER *Input*  
 4: 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.
- 5: 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  $k = k_1, \dots, k_2 - 1$ .
- 6: S(\*) – COMPLEX (KIND=nag\_wp) array *Input/Output*  
 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  $k = k_1, \dots, k_2 - 1$ .  
 On exit:  $S(k)$  holds a nonzero element of the spike of *H*:  $h_{k_2,k}$  if SIDE = 'L', or  $h_{k+1,k_1}$  if SIDE = 'R', for  $k = k_1, \dots, k_2 - 1$ .
- 7: 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  $n$  by  $n$  upper triangular matrix *U*. The imaginary parts of the diagonal elements must be zero.  
 On exit: the upper triangular part of the upper spiked matrix *H*. The imaginary parts of the diagonal elements are set to zero except for the  $(k_2, k_2)$  element if SIDE = 'L', or the  $(k_1, k_1)$  element if SIDE = 'R'.
- 8: LDA – INTEGER *Input*  
 On entry: the first dimension of the array *A* as declared in the (sub)program from which F06TWF is called.  
 Constraint:  $LDA \geq \max(1, N)$ .

## 6 Error Indicators and Warnings

None.

## 7 Accuracy

Not applicable.

## 8 Parallelism and Performance

F06TWF is not threaded in any implementation.

## **9 Further Comments**

None.

## **10 Example**

None.

---