

# NAG Fortran Library Routine Document

## F06ZWF (ZSYR2K)

**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

F06ZWF (ZSYR2K) performs one of the symmetric rank- $2k$  update operations

$$C \leftarrow \alpha AB^T + \alpha BA^T + \beta C \quad \text{or} \quad C \leftarrow \alpha A^T B + \alpha B^T A + \beta C,$$

where  $A$  and  $B$  are complex matrices,  $C$  is an  $n$  by  $n$  complex symmetric matrix, and  $\alpha$  and  $\beta$  are complex scalars.

### 2 Specification

```

SUBROUTINE F06ZWF (UPLO, TRANS, N, K, ALPHA, A, LDA, B, LDB, BETA, C,
1                      LDC)
    INTEGER           N, K, LDA, LDB, LDC
    complex*16      ALPHA, A(LDA,*), B(LDB,*), BETA, C(LDC,*)
    CHARACTER*1      UPLO, TRANS

```

The routine may be called by its BLAS name *zsyr2k*.

### 3 Description

None.

### 4 References

None.

### 5 Parameters

- 1: UPLO – CHARACTER\*1 *Input*  
*On entry:* specifies whether the upper or lower triangular part of  $C$  is stored as follows:  
 if UPLO = 'U', the upper triangular part of  $C$  is stored;  
 if UPLO = 'L', the lower triangular part of  $C$  is stored.  
*Constraint:* UPLO = 'U' or 'L'.
- 2: TRANS – CHARACTER\*1 *Input*  
*On entry:* specifies the operation to be performed as follows:  
 if TRANS = 'N',  $C \leftarrow \alpha AB^T + \alpha BA^T + \beta C$ ;  
 if TRANS = 'T',  $C \leftarrow \alpha A^T B + \alpha B^T A + \beta C$ .  
*Constraint:* TRANS = 'N' or 'T'.
- 3: N – INTEGER *Input*  
*On entry:*  $n$ , the order of the matrix  $C$ ; the number of rows of  $A$  and  $B$  if TRANS = 'N', or the number of columns of  $A$  and  $B$  otherwise.  
*Constraint:*  $N \geq 0$ .

- 4: K – INTEGER *Input*  
*On entry:*  $k$ , the number of columns of  $A$  and  $B$  if TRANS = 'N', or the number of rows of  $A$  and  $B$  otherwise.  
*Constraint:*  $K \geq 0$ .
- 5: ALPHA – *complex\*16* *Input*  
*On entry:* the scalar  $\alpha$ .
- 6: A(LDA,\*) – *complex\*16* array *Input*  
**Note:** the second dimension of the array  $A$  must be at least  $\max(1, K)$  if TRANS = 'N' and at least  $\max(1, N)$  otherwise.  
*On entry:* the matrix  $A$ ;  $A$  is  $n$  by  $k$  if TRANS = 'N', or  $k$  by  $n$  otherwise.
- 7: LDA – INTEGER *Input*  
*On entry:* the first dimension of the array  $A$  as declared in the (sub)program from which F06ZWF (ZSYR2K) is called.  
*Constraint:*  $LDA \geq \max(1, N)$  if TRANS = 'N';  $LDA \geq \max(1, K)$  otherwise.
- 8: B(LDB,\*) – *complex\*16* array *Input*  
**Note:** the second dimension of the array  $B$  must be at least  $\max(1, K)$  if TRANS = 'N' and at least  $\max(1, N)$  otherwise.  
*On entry:* the matrix  $B$ ;  $B$  is  $n$  by  $k$  if TRANS = 'N', or  $k$  by  $n$  otherwise.
- 9: LDB – INTEGER *Input*  
*On entry:* the first dimension of the array  $B$  as declared in the (sub)program from which F06ZWF (ZSYR2K) is called.  
*Constraint:*  $LDB \geq \max(1, N)$  if TRANS = 'N';  $LDB \geq \max(1, K)$  otherwise.
- 10: BETA – *complex\*16* *Input*  
*On entry:* the scalar  $\beta$ .
- 11: C(LDC,\*) – *complex\*16* array *Input/Output*  
**Note:** the second dimension of the array  $C$  must be at least  $\max(1, N)$ .  
*On entry:* the  $n$  by  $n$  symmetric matrix  $C$ . If UPLO = 'U', the upper triangle of  $C$  must be stored and the elements of the array below the diagonal are not referenced; if UPLO = 'L', the lower triangle of  $C$  must be stored and the elements of the array above the diagonal are not referenced.  
*On exit:* the updated matrix  $C$ .
- 12: LDC – INTEGER *Input*  
*On entry:* the first dimension of the array  $C$  as declared in the (sub)program from which F06ZWF (ZSYR2K) is called.  
*Constraint:*  $LDC \geq \max(1, N)$ .

## 6 Error Indicators and Warnings

None.

---