NAG Fortran Library Routine Document

F06ZRF (ZHER2K)

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

F06ZRF (ZHER2K) performs one of the Hermitian rank-2k update operations

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

where A and B are complex matrices, C is an n by n complex Hermitian matrix, α is a complex scalar, and β is a real scalar.

2 Specification

```
SUBROUTINE F06ZRF (UPLO, TRANS, N, K, ALPHA, A, LDA, B, LDB, BETA, C,
1                  LDC)
INTEGER            N, K, LDA, LDB, LDC
double precision   BETA
complex*16         ALPHA, A(LDA,*), B(LDB,*), C(LDC,*)
CHARACTER*1        UPLO, TRANS
```

The routine may be called by its BLAS name ***zher2k***.

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', $C \leftarrow \alpha AB^H + \bar{\alpha}BA^H + \beta C$;
if TRANS = 'C', $C \leftarrow \alpha A^H B + \bar{\alpha}B^H A + \beta C$.

Constraint: TRANS = 'N' or 'C'.

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 ≥ 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: $\mathrm{K} \geq 0$.

5: ALPHA – ***complex*16*** *Input*

On entry: the scalar α.

6: A(LDA,$*$) – ***complex*16*** array *Input*

Note: the second dimension of the array A must be at least $\max(1, \mathrm{K})$ if TRANS = 'N' and at least $\max(1, \mathrm{N})$ 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 F06ZRF (ZHER2K) is called.

Constraint: $\mathrm{LDA} \geq \max(1, \mathrm{N})$ if TRANS = 'N'; $\mathrm{LDA} \geq \max(1, \mathrm{K})$ otherwise.

8: B(LDB,$*$) – ***complex*16*** array *Input*

Note: the second dimension of the array B must be at least $\max(1, \mathrm{K})$ if TRANS = 'N' and at least $\max(1, \mathrm{N})$ 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 F06ZRF (ZHER2K) is called.

Constraint: $\mathrm{LDB} \geq \max(1, \mathrm{N})$ if TRANS = 'N'; $\mathrm{LDB} \geq \max(1, \mathrm{K})$ otherwise.

10: BETA – ***double precision*** *Input*

On entry: the scalar β.

11: C(LDC,$*$) – ***complex*16*** array *Input/Output*

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

On entry: the n by n Hermitian 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. The imaginary parts of the diagonal elements are set to zero.

12: LDC – INTEGER *Input*

On entry: the first dimension of the array C as declared in the (sub)program from which F06ZRF (ZHER2K) is called.

Constraint: $\mathrm{LDC} \geq \max(1, \mathrm{N})$.

6 Error Indicators and Warnings

None.