

NAG Library Routine Document

F06ZCF (ZHEMM)

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

F06ZCF (ZHEMM) performs one of the matrix-matrix operations

$$C \leftarrow \alpha AB + \beta C \quad \text{or} \quad C \leftarrow \alpha BA + \beta C,$$

where A is a complex Hermitian matrix, B and C are m by n complex matrices, and α and β are complex scalars.

2 Specification

```
SUBROUTINE F06ZCF (SIDE, UPLO, M, N, ALPHA, A, LDA, B, LDB, BETA, C,      &
                  LDC)
```

```
INTEGER                M, N, LDA, LDB, LDC
COMPLEX (KIND=nag_wp) ALPHA, A(LDA,*), B(LDB,*), BETA, C(LDC,*)
CHARACTER(1)          SIDE, UPLO
```

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

3 Description

None.

4 References

None.

5 Parameters

- 1: SIDE – CHARACTER(1) *Input*
On entry: specifies whether B is operated on from the left or the right.
 SIDE = 'L'
 B is pre-multiplied from the left.
 SIDE = 'R'
 B is post-multiplied from the right.
Constraint: SIDE = 'L' or 'R'.
- 2: UPLO – CHARACTER(1) *Input*
On entry: specifies whether the upper or lower triangular part of A is stored.
 UPLO = 'U'
 The upper triangular part of A is stored.
 UPLO = 'L'
 The lower triangular part of A is stored.
Constraint: UPLO = 'U' or 'L'.

- 3: M – INTEGER *Input*
On entry: m , the number of rows of the matrices B and C ; the order of A if SIDE = 'L'.
Constraint: $M \geq 0$.
- 4: N – INTEGER *Input*
On entry: n , the number of columns of the matrices B and C ; the order of A if SIDE = 'R'.
Constraint: $N \geq 0$.
- 5: ALPHA – COMPLEX (KIND=nag_wp) *Input*
On entry: the scalar α .
- 6: A(LDA,*) – COMPLEX (KIND=nag_wp) array *Input*
Note: the second dimension of the array A must be at least $\max(1, M)$ if SIDE = 'L' and at least $\max(1, N)$ if SIDE = 'R'.
On entry: the Hermitian matrix A ; A is m by m if SIDE = 'L', or n by n if SIDE = 'R'.
 If UPLO = 'U', the upper triangular part of A must be stored and the elements of the array below the diagonal are not referenced.
 If UPLO = 'L', the lower triangular part of A must be stored and the elements of the array above the diagonal are not referenced.
- 7: LDA – INTEGER *Input*
On entry: the first dimension of the array A as declared in the (sub)program from which F06ZCF (ZHEMM) is called.
Constraints:
 if SIDE = 'L', $LDA \geq \max(1, M)$;
 if SIDE = 'R', $LDA \geq \max(1, N)$.
- 8: B(LDB,*) – COMPLEX (KIND=nag_wp) array *Input*
Note: the second dimension of the array B must be at least $\max(1, N)$.
On entry: the m by n matrix B .
- 9: LDB – INTEGER *Input*
On entry: the first dimension of the array B as declared in the (sub)program from which F06ZCF (ZHEMM) is called.
Constraint: $LDB \geq \max(1, M)$.
- 10: BETA – COMPLEX (KIND=nag_wp) *Input*
On entry: the scalar β .
- 11: C(LDC,*) – COMPLEX (KIND=nag_wp) array *Input/Output*
Note: the second dimension of the array C must be at least $\max(1, N)$.
On entry: the m by n matrix C .
 If BETA = 0, C need not be set.
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 F06ZCF (ZHEMM) is called.

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

6 Error Indicators and Warnings

None.

7 Accuracy

Not applicable.

8 Parallelism and Performance

Not applicable.

9 Further Comments

None.

10 Example

None.
