NAG C Library Function Document

zhemm (f06zcc)

1 Purpose

zhemm (f06zcc) performs one of the matrix-matrix operations

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

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

```
#include <nag.h>
#include <nagf06.h>
void zhemm (OperationSide side, MatrixTriangle uplo, Integer m, Integer n,
      Complex alpha, const Complex a[], Integer tda, const Complex b[], Integer tdb,
      Complex beta, Complex c[], Integer tdc)
```

3 Arguments

1: **side** – OperationSide *Input*

On entry: specifies whether B is operated on from the left or the right, as follows:

if **side** = **LeftSide**, B is pre-multiplied from the left;
if **side** = **RightSide**, B is post-multiplied from the right.

Constraint: **side** = **LeftSide** or **RightSide**.

2: **uplo** – MatrixTriangle *Input*

On entry: specifies whether the upper or lower triangular part of A is stored as follows:

if **uplo** = **UpperTriangle**, the upper triangular part of A is stored;
if **uplo** = **LowerTriangle**, the lower triangular part of A is stored.

Constraint: **uplo** = **UpperTriangle** or **LowerTriangle**.

3: **m** – Integer *Input*

On entry: m, the number of rows of the matrices B and C; the order of A if **side** = **LeftSide**.

Constraint: **m** ≥ 0.

4: **n** – Integer *Input*

On entry: n, the number of columns of the matrices B and C; the order of A if **side** = **RightSide**.

Constraint: **n** ≥ 0.

5: **alpha** – Complex *Input*

On entry: the scalar α.

6: **a**[$\times$ **tda**] – const Complex *Input*

On entry: the Hermitian matrix A.

side = **LeftSide**

A is m by m.

side = **RightSide**

n by n.

uplo = **UpperTriangle**

The upper triangle of A must be stored and the elements of the array below the diagonal are not referenced.

uplo = **LowerTriangle**

The lower triangle of A must be stored and the elements of the array above the diagonal are not referenced.

7: **tda** – Integer *Input*

On entry: the second dimension of the array **a** as declared in the function from which zhemm (f06zcc) is called.

Constraint: **tda** $\geq max(1,$**m**$)$ if **side** = **LeftSide**; **tda** $\geq max(1,$**n**$)$ if **side** = **RightSide**.

8: **b**[**m** × **tdb**] – const Complex *Input*

On entry: the m by n matrix B.

9: **tdb** – Integer *Input*

On entry: the second dimension of the array **b** as declared in the function from which zhemm (f06zcc) is called.

Constraint: **tdb** $\geq \max(1,$**n**$)$.

10: **beta** – Complex *Input*

On entry: the scalar β.

11: **c**[**m** × **tdc**] – Complex *Input/Output*

On entry: the m by n matrix C. If **beta** = 0, **c** need not be set.

On exit: the updated matrix C.

12: **tdc** – Integer *Input*

On entry: the second dimension of the array **c** as declared in the function from which zhemm (f06zcc) is called.

Constraint: **tdc** $\geq \max(1,$**n**$)$.

4 Error Indicators and Warnings

If a function is called with an invalid argument then an error message is output on stderr, giving the name of the function and the number of the first invalid argument, and execution is terminated.