

NAG C Library Function Document

zgemm (f06zac)

1 Purpose

zgemm (f06zac) performs one of the matrix-matrix operations

$$\begin{array}{lll} C \leftarrow \alpha AB + \beta C, & C \leftarrow \alpha A^T B + \beta C, & C \leftarrow \alpha A^H B + \beta C, \\ C \leftarrow \alpha AB^T + \beta C, & C \leftarrow \alpha A^T B^T + \beta C, & C \leftarrow \alpha A^H B^T + \beta C, \\ C \leftarrow \alpha AB^H + \beta C, & C \leftarrow \alpha A^T B^H + \beta C & \text{or} & C \leftarrow \alpha A^H B^H + \beta C, \end{array}$$

where A , B and C are complex matrices, and α and β are complex scalars; C is always m by n .

2 Specification

```
#include <nag.h>
#include <nagf06.h>
```

```
void zgemm (MatrixTranspose transa, MatrixTranspose transb, Integer m, Integer n,
           Integer k, Complex alpha, const Complex a[], Integer tda, const Complex b[],
           Integer tdb, Complex beta, Complex c[], Integer tdc)
```

3 Arguments

1: **transa** – MatrixTranspose

Input

On entry: specifies whether the operation involves A , A^T or A^H , as follows:

- if **transa** = **NoTranspose**, it involves A ;
- if **transa** = **Transpose**, it involves A^T ;
- if **transa** = **ConjugateTranspose**, it involves A^H .

Constraint: **transa** = **NoTranspose**, **Transpose** or **ConjugateTranspose**.

2: **transb** – MatrixTranspose

Input

On entry: specifies whether the operation involves B , B^T or B^H , as follows:

- if **transb** = **NoTranspose**, it involves B ;
- if **transb** = **Transpose**, it involves B^T ;
- if **transb** = **ConjugateTranspose**, it involves B^H .

Constraint: **transb** = **NoTranspose**, **Transpose** or **ConjugateTranspose**.

3: **m** – Integer

Input

On entry: m , the number of rows of the matrix C ; the number of rows of A if **transa** = **NoTranspose**, or the number of columns of A if **transa** = **Transpose** or **ConjugateTranspose**.

Constraint: $m \geq 0$.

4: **n** – Integer

Input

On entry: n , the number of columns of the matrix C ; the number of columns of B if **transb** = **NoTranspose**, or the number of rows of B if **transb** = **Transpose** or **ConjugateTranspose**.

Constraint: $n \geq 0$.

- 5: **k** – Integer *Input*
On entry: k , the number of columns of A , if **transa** = **NoTranspose**, or the number of rows of A if **transa** = **Transpose** or **ConjugateTranspose**; the number of rows of B if **transb** = **NoTranspose**, or the number of columns of B if **transb** = **Transpose** or **ConjugateTranspose**.
Constraint: $k \geq 0$.
- 6: **alpha** – Complex *Input*
On entry: the scalar α .
- 7: **a**[\times **tda**] – const Complex *Input*
On entry: the matrix A ; A is m by k if **transa** = **NoTranspose**, or k by m if **transa** = **Transpose** or **ConjugateTranspose**.
- 8: **tda** – Integer *Input*
On entry: the second dimension of the array **a** as declared in the function from which `zgemm` (`f06zac`) is called.
Constraint: **tda** $\geq \max(1, k)$ if **transa** = **NoTranspose**; **tda** $\geq \max(1, m)$ if **transa** = **Transpose** or **ConjugateTranspose**.
- 9: **b**[\times **tdb**] – const Complex *Input*
On entry: the matrix B ; B is k by n if **transb** = **NoTranspose**, or n by k if **transb** = **Transpose** or **ConjugateTranspose**.
- 10: **tdb** – Integer *Input*
On entry: the second dimension of the array **b** as declared in the function from which `zgemm` (`f06zac`) is called.
Constraint: **tdb** $\geq \max(1, n)$ if **transb** = **NoTranspose**; **tdb** $\geq \max(1, k)$ if **transb** = **Transpose** or **ConjugateTranspose**.
- 11: **beta** – Complex *Input*
On entry: the scalar β .
- 12: **c**[$m \times$ **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 .
- 13: **tdc** – Integer *Input*
On entry: the second dimension of the array **c** as declared in the function from which `zgemm` (`f06zac`) 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.