

# NAG C Library Function Document

## ztrsm (f06zjc)

### 1 Purpose

ztrsm (f06zjc) performs one of the matrix-matrix operations

$$\begin{aligned} B &\leftarrow \alpha A^{-1}B, & B &\leftarrow \alpha A^{-T}B, & B &\leftarrow \alpha A^{-H}B, \\ B &\leftarrow \alpha BA^{-1}, & B &\leftarrow \alpha BA^{-T} & \text{or} & B &\leftarrow \alpha BA^{-H}, \end{aligned}$$

where  $A$  is a complex triangular matrix,  $B$  is an  $m$  by  $n$  complex matrix, and  $\alpha$  is a complex scalar.  $A^{-T}$  denotes  $(A^T)^{-1}$  or equivalently  $(A^{-1})^T$ ;  $A^{-H}$  denotes  $(A^H)^{-1}$  or equivalently  $(A^{-1})^H$ .

No test for singularity or near-singularity of  $A$  is included in this function. Such tests must be performed before calling this function.

### 2 Specification

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

```
void ztrsm (OperationSide side, MatrixTriangle uplo, MatrixTranspose transa,
           MatrixUnitTriangular diag, Integer m, Integer n, Complex alpha,
           const Complex a[], Integer tda, Complex b[], Integer tdb)
```

### 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  $A$  upper or lower triangular as follows:  
     if **uplo** = **UpperTriangle**,  $A$  is upper triangular;  
     if **uplo** = **LowerTriangle**,  $A$  is lower triangular.  
*Constraint:* **uplo** = **UpperTriangle** or **LowerTriangle**.
- 3: **transa** – MatrixTranspose *Input*  
*On entry:* specifies whether the operation involves  $A^{-1}$ ,  $A^{-T}$  or  $A^{-H}$ , as follows:  
     if **transa** = **NoTranspose**, it involves  $A^{-1}$ ;  
     if **transa** = **Transpose**, it involves  $A^{-T}$ ;  
     if **transa** = **ConjugateTranspose**, it involves  $A^{-H}$ .  
*Constraint:* **transa** = **NoTranspose**, **Transpose** or **ConjugateTranspose**.

- 4: **diag** – MatrixUnitTriangular *Input*  
*On entry:* specifies whether  $A$  has non-unit or unit diagonal elements, as follows:  
 if **diag** = **NotUnitTriangular**, the diagonal elements are stored explicitly;  
 if **diag** = **UnitTriangular**, the diagonal elements are assumed to be 1, and are not referenced.  
*Constraint:* **diag** = **NotUnitTriangular** or **UnitTriangular**.
- 5: **m** – Integer *Input*  
*On entry:*  $m$ , the number of rows of the matrix  $B$ ; the order of  $A$  if **side** = **LeftSide**.  
*Constraint:*  $m \geq 0$ .
- 6: **n** – Integer *Input*  
*On entry:*  $n$ , the number of columns of the matrix  $B$ ; the order of  $A$  if **side** = **RightSide**.  
*Constraint:*  $n \geq 0$ .
- 7: **alpha** – Complex *Input*  
*On entry:* the scalar  $\alpha$ .
- 8: **a**[ $m \times tda$ ] – const Complex *Input*  
*On entry:* the triangular matrix  $A$ .  
 if **side** = **LeftSide**,  $A$  is  $m$  by  $m$ .  
**side** = **RightSide**  
 $n$  by  $n$ .  
**uplo** = **UpperTriangle**  
 $A$  is upper triangular and the elements of the array below the diagonal are not referenced.  
**uplo** = **LowerTriangle**  
 $A$  is lower triangular and the elements of the array above the diagonal are not referenced.  
**diag** = **UnitTriangular**  
 The diagonal elements of  $A$  are not referenced, but are assumed to be 1.
- 9: **tda** – Integer *Input*  
*On entry:* the second dimension of the array **a** as declared in the function from which `ztrsm` (f06zjc) is called.  
*Constraint:* **tda**  $\geq \max(1, m)$  if **side** = **LeftSide**; **tda**  $\geq \max(1, n)$  if **side** = **RightSide**.
- 10: **b**[ $m \times tdb$ ] – Complex *Input/Output*  
*On entry:* the  $m$  by  $n$  matrix  $B$ . If **alpha** = 0, **b** need not be set.  
*On exit:* the updated matrix  $B$ .
- 11: **tdb** – Integer *Input*  
*On entry:* the second dimension of the array **b** as declared in the function from which `ztrsm` (f06zjc) is called.  
*Constraint:* **tdb**  $\geq \max(1, n)$ .

## **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.

---