

# NAG Fortran Library Routine Document

## F06TAF (ZSYMV)

**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

F06TAF (ZSYMV) performs the matrix-vector operation

$$y \leftarrow \alpha Ax + \beta y$$

where  $A$  is an  $n$  by  $n$  complex symmetric matrix,  $x$  and  $y$  are  $n$  element complex vectors, and  $\alpha$  and  $\beta$  are complex scalars.

### 2 Specification

```
SUBROUTINE F06TAF (UPLO, N, ALPHA, A, LDA, X, INCX, BETA, Y, INCY)
INTEGER          N, LDA, INCX, INCY
complex*16      ALPHA, A(LDA,*), X(*), BETA, Y(*)
CHARACTER*1      UPLO
```

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

### 3 Description

None.

### 4 References

None.

### 5 Parameters

- 1: UPLO – CHARACTER\*1 *Input*  
*On entry:* specifies whether the upper or lower triangular part of  $A$  is stored as follows:  
     if UPLO = 'U', the upper triangular part of  $A$  is stored;  
     if UPLO = 'L', the lower triangular part of  $A$  is stored.  
*Constraint:* UPLO = 'U' or 'L'.
- 2: N – INTEGER *Input*  
*On entry:*  $n$ , the order of the matrix  $A$ .  
*Constraint:*  $N \geq 0$ .
- 3: ALPHA – ***complex\*16*** *Input*  
*On entry:* the scalar  $\alpha$ .
- 4: A(LDA,\*) – ***complex\*16*** array *Input*  
**Note:** the second dimension of the array  $A$  must be at least  $\max(1, N)$ .  
*On entry:* the  $n$  by  $n$  symmetric matrix  $A$ . If UPLO = 'U', the upper triangle of  $A$  must be stored and the elements of the array below the diagonal are not referenced; if UPLO = 'L', the lower triangle of  $A$  must be stored and the elements of the array above the diagonal are not referenced.

- 5: LDA – INTEGER *Input*  
*On entry:* the first dimension of the array A as declared in the (sub)program from which F06TAF (ZSYMV) is called.  
*Constraint:*  $LDA \geq \max(1, N)$ .
- 6: X(\*) – **complex\*16** array *Input*  
*On entry:* the vector  $x$ .
- 7: INCX – INTEGER *Input*  
*On entry:* the increment in the subscripts of X between successive elements of  $x$ .  
*Constraint:*  $INCX \neq 0$ .
- 8: BETA – **complex\*16** *Input*  
*On entry:* the scalar  $\beta$ .
- 9: Y(\*) – **complex\*16** array *Input/Output*  
*On entry:* the vector  $y$ . If BETA = 0, Y need not be set.  
*On exit:* the updated vector  $y$ .
- 10: INCY – INTEGER *Input*  
*On entry:* the increment in the subscripts of Y between successive elements of  $y$ .  
*Constraint:*  $INCY \neq 0$ .

## 6 Error Indicators and Warnings

None.

---