

NAG Library Routine Document

F06SHF (ZTPMV)

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

F06SHF (ZTPMV) computes the matrix-vector product for a complex triangular matrix, its transpose or its conjugate transpose, stored in packed form.

2 Specification

```
SUBROUTINE F06SHF (UPLO, TRANS, DIAG, N, AP, X, INCX)
  INTEGER          N, INCX
  COMPLEX (KIND=nag_wp) AP(*), X(*)
  CHARACTER(1)    UPLO, TRANS, DIAG
```

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

3 Description

F06SHF (ZTPMV) performs one of the matrix-vector operations

$$x \leftarrow Ax, \quad x \leftarrow A^T x \quad \text{or} \quad x \leftarrow A^H x,$$

where A is an n by n complex triangular matrix, stored in packed form, and x is an n -element complex vector.

4 References

None.

5 Parameters

- 1: UPLO – CHARACTER(1) *Input*
On entry: specifies whether A is upper or lower triangular.
 UPLO = 'U'
 A is upper triangular.
 UPLO = 'L'
 A is lower triangular.
Constraint: UPLO = 'U' or 'L'.
- 2: TRANS – CHARACTER(1) *Input*
On entry: specifies the operation to be performed.
 TRANS = 'N'
 $x \leftarrow Ax$.
 TRANS = 'T'
 $x \leftarrow A^T x$.
 TRANS = 'C'
 $x \leftarrow A^H x$.
Constraint: TRANS = 'N', 'T' or 'C'.

- 3: **DIAG** – CHARACTER(1) *Input*
On entry: specifies whether A has nonunit or unit diagonal elements.
DIAG = 'N'
 The diagonal elements are stored explicitly.
DIAG = 'U'
 The diagonal elements are assumed to be 1, and are not referenced.
Constraint: **DIAG** = 'N' or 'U'.
- 4: **N** – INTEGER *Input*
On entry: n , the order of the matrix A .
Constraint: $N \geq 0$.
- 5: **AP(*)** – COMPLEX (KIND=nag_wp) array *Input*
Note: the dimension of the array **AP** must be at least $N \times (N + 1)/2$.
On entry: the n by n triangular matrix A , packed by columns.
 More precisely,
 if **UPLO** = 'U', the upper triangle of A must be stored with element A_{ij} in
 AP($i + j(j - 1)/2$) for $i \leq j$;
 if **UPLO** = 'L', the lower triangle of A must be stored with element A_{ij} in
 AP($i + (2n - j)(j - 1)/2$) for $i \geq j$.
 If **DIAG** = 'U', the diagonal elements of A are assumed to be 1, and are not referenced; the same storage scheme is used whether **DIAG** = 'N' or 'U'.
- 6: **X(*)** – COMPLEX (KIND=nag_wp) array *Input/Output*
Note: the dimension of the array **X** must be at least $\max(1, 1 + (N - 1) \times |\text{INCX}|)$.
On entry: the vector x .
 If **INCX** > 0, x_i must be stored in **X**($1 + (i - 1) \times \text{INCX}$), for $i = 1, 2, \dots, N$.
 If **INCX** < 0, x_i must be stored in **X**($1 - (N - i) \times \text{INCX}$), for $i = 1, 2, \dots, N$.
On exit: the updated vector x stored in the array elements used to supply the original vector x .
- 7: **INCX** – INTEGER *Input*
On entry: the increment in the subscripts of **X** between successive elements of x .
Constraint: **INCX** $\neq 0$.

6 Error Indicators and Warnings

None.

7 Accuracy

Not applicable.

8 Parallelism and Performance

Not applicable.

9 Further Comments

None.

10 Example

None.
