

NAG Library Routine Document

F06PEF (DSPMV)

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

F06PEF (DSPMV) computes the matrix-vector product for a real symmetric matrix stored in packed form.

2 Specification

```
SUBROUTINE F06PEF (UPLO, N, ALPHA, AP, X, INCX, BETA, Y, INCY)
```

```
INTEGER          N, INCX, INCY
REAL (KIND=nag_wp) ALPHA, AP(*), X(*), BETA, Y(*)
CHARACTER(1)     UPLO
```

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

3 Description

F06PEF (DSPMV) performs the matrix-vector operation

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

where A is an n by n real symmetric matrix stored in packed form, x and y are n -element real vectors, and α and β are real scalars.

4 References

None.

5 Parameters

- | | | |
|----|---|--------------|
| 1: | UPLO – CHARACTER(1) | <i>Input</i> |
| | <i>On entry:</i> specifies whether the upper or lower triangular part of A is stored. | |
| | UPLO = 'U'
The upper triangular part of A is stored. | |
| | UPLO = 'L'
The lower triangular part of A is stored. | |
| | <i>Constraint:</i> UPLO = 'U' or 'L'. | |
| 2: | N – INTEGER | <i>Input</i> |
| | <i>On entry:</i> n , the order of the matrix A . | |
| | <i>Constraint:</i> $N \geq 0$. | |
| 3: | ALPHA – REAL (KIND=nag_wp) | <i>Input</i> |
| | <i>On entry:</i> the scalar α . | |

- 4: AP(*) – REAL (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 symmetric 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 $i \geq j$.
- 5: X(*) – REAL (KIND=nag_wp) array Input
Note: the dimension of the array X must be at least $\max(1, 1 + (N - 1) \times |\text{INCX}|)$.
On entry: the n -element 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 $i = 1, 2, \dots, N$.
 Intermediate elements of X are not referenced.
- 6: INCX – INTEGER Input
On entry: the increment in the subscripts of X between successive elements of x .
Constraint: INCX \neq 0.
- 7: BETA – REAL (KIND=nag_wp) Input
On entry: the scalar β .
- 8: Y(*) – REAL (KIND=nag_wp) array Input/Output
Note: the dimension of the array Y must be at least $\max(1, 1 + (N - 1) \times |\text{INCY}|)$.
On entry: the n -element vector y , if BETA = 0.0 Y need not be set.
 If INCY > 0, y_i must be stored in Y($1 + (i - 1) \times \text{INCY}$), for $i = 1, 2, \dots, N$.
 If INCY < 0, y_i must be stored in Y($1 - (N - i) \times \text{INCY}$), for $i = 1, 2, \dots, N$.
On exit: the updated vector y stored in the array elements used to supply the original vector y .
- 9: 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.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

None.
