

# NAG Library Routine Document

## F06PSF (DSPR2)

**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

F06PSF (DSPR2) computes the rank-2 update of a real symmetric matrix stored in packed form.

### 2 Specification

```
SUBROUTINE F06PSF (UPLO, N, ALPHA, X, INCX, Y, INCY, AP)
  INTEGER          N, INCX, INCY
  REAL (KIND=nag_wp) ALPHA, X(*), Y(*), AP(*)
  CHARACTER(1)    UPLO
```

The routine may be called by its BLAS name *dspr2*.

### 3 Description

F06PSF (DSPR2) performs the symmetric rank-2 update operation

$$A \leftarrow \alpha xy^T + \alpha yx^T + A,$$

where  $A$  is an  $n$  by  $n$  real symmetric matrix, stored in packed form,  $x$  and  $y$  are  $n$ -element real vectors, and  $\alpha$  is a real scalar.

### 4 References

None.

### 5 Arguments

- |    |                                                                                                          |              |
|----|----------------------------------------------------------------------------------------------------------|--------------|
| 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'<br>The upper triangular part of $A$ is stored.                                                |              |
|    | UPLO = 'L'<br>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 $\alpha$ .                                                                   |              |
| 4: | X(*) – REAL (KIND=nag_wp) array                                                                          | <i>Input</i> |
|    | <b>Note:</b> the dimension of the array X must be at least $\max(1, 1 + (N - 1) \times  \text{INCX} )$ . |              |
|    | <i>On entry:</i> the $n$ -element vector $x$ .                                                           |              |

If  $\text{INCX} > 0$ ,  $x_i$  must be stored in  $X(1 + (i - 1) \times \text{INCX})$ , for  $i = 1, 2, \dots, N$ .

If  $\text{INCX} < 0$ ,  $x_i$  must be stored in  $X(1 - (N - i) \times \text{INCX})$ , for  $i = 1, 2, \dots, N$ .

Intermediate elements of  $X$  are not referenced.

- 5:  $\text{INCX}$  – INTEGER *Input*  
*On entry:* the increment in the subscripts of  $X$  between successive elements of  $x$ .  
*Constraint:*  $\text{INCX} \neq 0$ .
- 6:  $Y(*)$  – REAL (KIND=nag\_wp) array *Input*  
**Note:** the dimension of the array  $Y$  must be at least  $\max(1, 1 + (N - 1) \times |\text{INCX}|)$ .  
*On entry:* the  $n$ -element vector  $y$ .  
 If  $\text{INCX} > 0$ ,  $y_i$  must be stored in  $Y(1 + (i - 1) \times \text{INCX})$ , for  $i = 1, 2, \dots, N$ .  
 If  $\text{INCX} < 0$ ,  $y_i$  must be stored in  $Y(1 - (N - i) \times \text{INCX})$ , for  $i = 1, 2, \dots, N$ .  
 Intermediate elements of  $Y$  are not referenced.
- 7:  $\text{INCX}$  – INTEGER *Input*  
*On entry:* the increment in the subscripts of  $Y$  between successive elements of  $y$ .  
*Constraint:*  $\text{INCX} \neq 0$ .
- 8:  $\text{AP}(*)$  – REAL (KIND=nag\_wp) array *Input/Output*  
**Note:** the dimension of the array  $\text{AP}$  must be at least  $N \times (N + 1)/2$ .  
*On entry:* the  $n$  by  $n$  symmetric matrix  $A$ , packed by columns.  
 More precisely,  
     if  $\text{UPLO} = 'U'$ , the upper triangle of  $A$  must be stored with element  $A_{ij}$  in  $\text{AP}(i + j(j - 1)/2)$  for  $i \leq j$ ;  
     if  $\text{UPLO} = 'L'$ , the lower triangle of  $A$  must be stored with element  $A_{ij}$  in  $\text{AP}(i + (2n - j)(j - 1)/2)$  for  $i \geq j$ .  
*On exit:* the updated matrix  $A$ .

## 6 Error Indicators and Warnings

None.

## 7 Accuracy

Not applicable.

## 8 Parallelism and Performance

F06PSF (DSPR2) is not threaded in any implementation.

## 9 Further Comments

None.

## **10 Example**

None.

---