

# NAG Library Routine Document

## F06SBF (ZGBMV)

**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

F06SBF (ZGBMV) computes the matrix-vector product for a complex general band matrix, its transpose or its conjugate transpose.

### 2 Specification

SUBROUTINE F06SBF (TRANS, M, N, KL, KU, ALPHA, A, LDA, X, INCX, BETA, Y, &  
INCY)

INTEGER M, N, KL, KU, LDA, INCX, INCY  
COMPLEX (KIND=nag\_wp) ALPHA, A(LDA,\*), X(\*), BETA, Y(\*)  
CHARACTER(1) TRANS

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

### 3 Description

F06SBF (ZGBMV) performs one of the matrix-vector operations

$$y \leftarrow \alpha Ax + \beta y, \quad y \leftarrow \alpha A^T x + \beta y \quad \text{or} \quad y \leftarrow \alpha A^H x + \beta y,$$

where  $A$  is an  $m$  by  $n$  complex band matrix with  $k_l$  subdiagonals and  $k_u$  superdiagonals,  $x$  and  $y$  are complex vectors, and  $\alpha$  and  $\beta$  are complex scalars.

If  $m = 0$  or  $n = 0$ , no operation is performed.

### 4 References

None.

### 5 Arguments

1: TRANS – CHARACTER(1) *Input*

*On entry:* specifies the operation to be performed.

TRANS = 'N'

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

TRANS = 'T'

$$y \leftarrow \alpha A^T x + \beta y.$$

TRANS = 'C'

$$y \leftarrow \alpha A^H x + \beta y.$$

*Constraint:* TRANS = 'N', 'T' or 'C'.

2: M – INTEGER *Input*

*On entry:*  $m$ , the number of rows of the matrix  $A$ .

*Constraint:*  $M \geq 0$ .

- 3: N – INTEGER *Input*  
*On entry:*  $n$ , the number of columns of the matrix  $A$ .  
*Constraint:*  $N \geq 0$ .
- 4: KL – INTEGER *Input*  
*On entry:*  $k_l$ , the number of subdiagonals within the band of  $A$ .  
*Constraint:*  $KL \geq 0$ .
- 5: KU – INTEGER *Input*  
*On entry:*  $k_u$ , the number of superdiagonals within the band of  $A$ .  
*Constraint:*  $KU \geq 0$ .
- 6: ALPHA – COMPLEX (KIND=nag\_wp) *Input*  
*On entry:* the scalar  $\alpha$ .
- 7: A(LDA,\*) – COMPLEX (KIND=nag\_wp) array *Input*  
**Note:** the second dimension of the array  $A$  must be at least  $N$ .  
*On entry:* the  $m$  by  $n$  band matrix  $A$ .  
The matrix is stored in rows 1 to  $k_l + k_u + 1$ , more precisely, the element  $A_{ij}$  must be stored in  

$$A(k_u + 1 + i - j, j) \quad \text{for } \max(1, j - k_u) \leq i \leq \min(m, j + k_l).$$
- 8: LDA – INTEGER *Input*  
*On entry:* the first dimension of the array  $A$  as declared in the (sub)program from which F06SBF (ZGBMV) is called.  
*Constraint:*  $LDA \geq KL + KU + 1$ .
- 9: X(\*) – COMPLEX (KIND=nag\_wp) array *Input*  
**Note:** the dimension of the array  $X$  must be at least  $\max(1, 1 + (N - 1) \times |\text{INCX}|)$  if  $\text{TRANS} = 'N'$  and at least  $\max(1, 1 + (M - 1) \times |\text{INCX}|)$  if  $\text{TRANS} = 'T'$  or  $'C'$ .  
*On entry:* the vector  $x$ .  
If  $\text{TRANS} = 'N'$ ,  
    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  $i = 1, 2, \dots, N$ .  
If  $\text{TRANS} = 'T'$  or  $'C'$ ,  
    if  $\text{INCX} > 0$ ,  $x_i$  must be stored in  $X(1 + (i - 1) \times \text{INCX})$ , for  $i = 1, 2, \dots, M$ ;  
    if  $\text{INCX} < 0$ ,  $x_i$  must be stored in  $X(1 - (M - i) \times \text{INCX})$ , for  $i = 1, 2, \dots, M$ .
- 10: INCX – INTEGER *Input*  
*On entry:* the increment in the subscripts of  $X$  between successive elements of  $x$ .  
*Constraint:*  $\text{INCX} \neq 0$ .
- 11: BETA – COMPLEX (KIND=nag\_wp) *Input*  
*On entry:* the scalar  $\beta$ .

12: Y(\*) – COMPLEX (KIND=nag\_wp) array *Input/Output*

**Note:** the dimension of the array Y must be at least  $\max(1, 1 + (M - 1) \times |\text{INCY}|)$  if TRANS = 'N' and at least  $\max(1, 1 + (N - 1) \times |\text{INCY}|)$  if TRANS = 'T' or 'C'.

*On entry:* the vector  $y$ , if BETA = 0.0, Y need not be set.

If TRANS = 'N',

if INCY > 0,  $y_i$  must be stored in  $Y(1 + (i - 1) \times \text{INCY})$ , for  $i = 1, 2, \dots, M$ ;

if INCY < 0,  $y_i$  must be stored in  $Y(1 - (M - i) \times \text{INCY})$ , for  $i = 1, 2, \dots, M$ .

If TRANS = 'T' or 'C',

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  $y$ .

13: 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 Parallelism and Performance

F06SBF (ZGBMV) is not threaded in any implementation.

## 9 Further Comments

None.

## 10 Example

None.

---