NAG Fortran 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) 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 sub-diagonals and k_u super-diagonals, x and y are complex vectors, and α and β are complex scalars.

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

2 Specification

```
SUBROUTINE F06SBF (TRANS, M, N, KL, KU, ALPHA, A, LDA, X, INCX, BETA, Y,
1                  INCY)
INTEGER            M, N, KL, KU, LDA, INCX, INCY
complex*16         ALPHA, A(LDA,*), X(*), BETA, Y(*)
CHARACTER*1        TRANS
```

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

3 Description

None.

4 References

None.

5 Parameters

1: TRANS – CHARACTER*1 *Input*

On entry: specifies the operation to be performed as follows:

if TRANS = 'N', $y \leftarrow \alpha Ax + \beta y$;
if TRANS = 'T', $y \leftarrow \alpha A^T x + \beta y$;
if 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: $\mathrm{M} \geq 0$.

3: N – INTEGER *Input*

On entry: n, the number of columns of the matrix A.

Constraint: $\mathrm{N} \geq 0$.

4: KL – INTEGER *Input*

On entry: k_l, the number of sub-diagonals within the band of A.

Constraint: $\mathrm{KL} \geq 0$.

5: KU – INTEGER *Input*

On entry: k_u, the number of super-diagonals within the band of A.

Constraint: $\mathrm{KU} \geq 0$.

6: ALPHA – ***complex*16*** *Input*

On entry: the scalar α.

7: A(LDA,$*$) – ***complex*16*** array *Input*

Note: the second dimension of the array A must be at least $\max(1, \mathrm{N})$.

On entry: the m by n band matrix A, stored in rows 1 to $k_l + k_u + 1$. More precisely, a_{ij} must be stored in $\mathrm{A}(k_u + i - j + 1, j)$ for $\max(j - k_u, 1) \leq i \leq \min(j + k_l, m)$.

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: $\mathrm{LDA} \geq \mathrm{KL} + \mathrm{KU} + 1$.

9: X($*$) – ***complex*16*** array *Input*

On entry: the vector x, of length n if TRANS = 'N', or of length m if TRANS = 'T' or 'C'.

10: INCX – INTEGER *Input*

On entry: the increment in the subscripts of X between successive elements of x.

Constraint: $\mathrm{INCX} \neq 0$.

11: BETA – ***complex*16*** *Input*

On entry: the scalar β.

12: Y($*$) – ***complex*16*** array *Input/Output*

On entry: the vector y, of length m if TRANS = 'N', or of length n if TRANS = 'T' or 'C'. If BETA = 0, Y need not be set.

On exit: the updated vector y.

13: INCY – INTEGER *Input*

On entry: the increment in the subscripts of Y between successive elements of y.

Constraint: $\mathrm{INCY} \neq 0$.

6 Error Indicators and Warnings

None.