NAG C Library Function Document

nag_dgb_norm (f16rbc)

1 Purpose
nag_dgb_norm (f16rbc) calculates the value of the 1-norm, the infinity-norm, the Frobenius norm, or the maximum absolute value of the elements, of a real m by n band matrix.

2 Specification
void nag_dgb_norm (Nag_OrderType order, Nag_NormType norm, Integer m, Integer n, Integer kl, Integer ku, const double ab[], Integer pdab, double *r, NagError *fail)

3 Description
Given a real m by n band matrix, A, nag_dgb_norm (f16rbc) calculates one of the values given by

\[ \|A\|_1 = \max_j \sum_{i=1}^{m} |a_{ij}|, \]

\[ \|A\|_\infty = \max_i \sum_{j=1}^{n} |a_{ij}|, \]

\[ \|A\|_F = \left( \sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2 \right)^{1/2}, \]

\[ \max_{i,j} |a_{ij}|. \]

4 References

5 Parameters
1: order – Nag_OrderType

On entry: the order parameter specifies the two-dimensional storage scheme being used, i.e., row-major ordering or column-major ordering. C language defined storage is specified by order = Nag_RowMajor. See Section 2.2.1.4 of the Essential Introduction for a more detailed explanation of the use of this parameter.

Constraint: order = Nag_RowMajor or Nag_ColMajor.

2: norm – Nag_NormType

On entry: specifies the value to be returned:

if norm = Nag_OneNorm, the 1-norm;
if norm = Nag_Infnorm, the infinity-norm;
if norm = Nag_FrobeniusNorm, the Frobenius (or Euclidean) norm;
if norm = Nag_MaxNorm, the value max \( i,j |a_{ij}| \) (not a norm).

Constraint: norm = Nag_OneNorm, Nag_Infnorm, Nag_FrobeniusNorm or Nag_MaxNorm.
3: \( m \) – Integer \hspace{1cm} \text{Input}

On entry: \( m \), the number of rows of the matrix \( A \).
Constraint: \( m \geq 0 \).

4: \( n \) – Integer \hspace{1cm} \text{Input}

On entry: \( n \), the number of columns of the matrix \( A \).
Constraint: \( n \geq 0 \).

5: \( k_l \) – Integer \hspace{1cm} \text{Input}

On entry: \( k_l \), the number of sub-diagonals within the band of \( A \).
Constraint: \( k_l \geq 0 \).

6: \( k_u \) – Integer \hspace{1cm} \text{Input}

On entry: \( k_u \), the number of super-diagonals within the band of \( A \).
Constraint: \( k_u \geq 0 \).

7: \( ab \) \[\text{dim}\] – const double \hspace{1cm} \text{Input}

Note: the dimension, \( \text{dim} \), of the array \( ab \) must be at least \( \max(1,pdab \times n) \) when \( \text{order} = \text{Nag} \_\text{ColMajor} \) and at least \( \max(1,pdab \times m) \) when \( \text{order} = \text{Nag} \_\text{RowMajor} \).

On entry: the \( m \) by \( n \) matrix \( A \). This is stored as a notional two-dimensional array with row elements or column elements stored contiguously. The storage of elements \( a_{ij} \), for \( i = 1, \ldots, m \) and \( j = \max(1,i - k_l), \ldots, \min(n,i + k_u) \), depends on the \text{order} parameter as follows:
- if \( \text{order} = \text{Nag} \_\text{ColMajor} \), \( a_{ij} \) is stored as \( ab[(j - 1) \times pdab + kl + ku + i - j] \);
- if \( \text{order} = \text{Nag} \_\text{RowMajor} \), \( a_{ij} \) is stored as \( ab[(i - 1) \times pdab + kl + j - i] \).

8: \( pdab \) – Integer \hspace{1cm} \text{Input}

On entry: the stride separating row or column elements (depending on the value of \text{order}) of the matrix \( A \) in the array \( ab \).
Constraint: \( pdab \geq kl + ku + 1 \).

9: \( r \) – double * \hspace{1cm} \text{Output}

On exit: the value of the norm specified by \text{norm}.

10: \( \text{fail} \) – NagError * \hspace{1cm} \text{Input/Output}

The NAG error parameter (see the Essential Introduction).

### 6 Error Indicators and Warnings

**NE_INT**

On entry, \( m = \langle \text{value} \rangle \).
Constraint: \( m \geq 0 \).

On entry, \( n = \langle \text{value} \rangle \).
Constraint: \( n \geq 0 \).

On entry, \( k_l = \langle \text{value} \rangle \).
Constraint: \( k_l \geq 0 \).

On entry, \( k_u = \langle \text{value} \rangle \).
Constraint: \( k_u \geq 0 \).
On entry, \( \text{pdab} = \langle \text{value} \rangle \).
Constraint: \( \text{pdab} \geq \text{kl} + \text{ku} + 1 \).

**NE_BAD_PARAM**
On entry, parameter \( \langle \text{value} \rangle \) had an illegal value.

### 7 Accuracy
The BLAS standard requires accurate implementations which avoid unnecessary over/underflow (see section 2.7 of The BLAS Technical Forum Standard (2001)).

### 8 Further Comments
None.

### 9 Example
None.