NAG C Library Function Document

nag_zgebak (f08nwc)

1 Purpose

nag_zgebak (f08nwc) transforms eigenvectors of a balanced matrix to those of the original complex general matrix.

2 Specification

void nag_zgebak (Nag_OrderType order, Nag_JobType job, Nag_SideType side, Integer n, Integer ilo, Integer ihi, const double scale[], Integer m, Complex v[], Integer pdv, NagError *fail)

3 Description

nag_zgebak (f08nwc) is intended to be used after a complex general matrix \( A \) has been balanced by nag_zgebal (f08nvc), and eigenvectors of the balanced matrix \( A_0^0 \) have subsequently been computed.

For a description of balancing, see the document for nag_zgebal (f08nvc). The balanced matrix \( A'' \) is obtained as \( A'' = DPAP^T D^{-1} \), where \( P \) is a permutation matrix and \( D \) is a diagonal scaling matrix. This function transforms left or right eigenvectors as follows:

- If \( x \) is a right eigenvector of \( A'' \), \( P^T D^{-1} x \) is a right eigenvector of \( A \);
- If \( y \) is a left eigenvector of \( A'' \), \( P^T D y \) is a left eigenvector of \( A \);

4 References

None.

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: job – Nag_JobType

   On entry: this must be the same parameter job as supplied to nag_zgebal (f08nvc).

   Constraint: job = Nag_DoNothing, Nag_Permute, Nag_Scale or Nag_DoBoth.

3: side – Nag_SideType

   On entry: indicates whether left or right eigenvectors are to be transformed, as follows:

   - If side = Nag_LeftSide, left eigenvectors are transformed;
   - If side = Nag_RightSide, right eigenvectors are transformed.

   Constraint: side = Nag_LeftSide or Nag_RightSide.
4: \( n \) – Integer
   \textit{Input}
   
   On entry: \( n \), the number of rows of the matrix of eigenvectors.
   
   Constraint: \( n \geq 0 \).

5: \( \text{ilo} \) – Integer
6: \( \text{ihi} \) – Integer
   \textit{Input}
   
   On entry: the values \( \text{ilo} \) and \( \text{ihi} \), as returned by nag_zgebal (f08nvc).
   
   Constraints:
   
   \[
   \begin{align*}
   \text{if } n > 0, & 1 \leq \text{ilo} \leq \text{ihi} \leq n; \\
   \text{if } n = 0, & \text{ilo} = 1 \text{ and } \text{ihi} = 0.
   \end{align*}
   \]

7: \( \text{scale}[\text{dim}] \) – const double
   \textit{Input}
   
   Note: the dimension, \( \text{dim} \), of the array \( \text{scale} \) must be at least \( \max(1,n) \).
   
   On entry: details of the permutations and/or the scaling factors used to balance the original complex general matrix, as returned by nag_zgebal (f08nvc).

8: \( m \) – Integer
   \textit{Input}
   
   On entry: \( m \), the number of columns of the matrix of eigenvectors.
   
   Constraint: \( m \geq 0 \).

9: \( \text{v}[\text{dim}] \) – Complex
   \textit{Input/Output}
   
   Note: the dimension, \( \text{dim} \), of the array \( \text{v} \) must be at least \( \max(1,pdv \times m) \) when \( \text{order} = \text{Nag\_ColMajor} \) and at least \( \max(1,pdv \times n) \) when \( \text{order} = \text{Nag\_RowMajor} \).
   
   If \( \text{order} = \text{Nag\_ColMajor} \), the \((i,j)\)th element of the matrix \( V \) is stored in \( \text{v}[(j-1) \times pdv + i - 1] \) and if \( \text{order} = \text{Nag\_RowMajor} \), the \((i,j)\)th element of the matrix \( V \) is stored in \( \text{v}[(i-1) \times pdv + j - 1] \).
   
   On entry: the matrix of left or right eigenvectors to be transformed.
   
   On exit: the transformed eigenvectors.

10: \( pdv \) – Integer
    \textit{Input}
    
    On entry: the stride separating matrix row or column elements (depending on the value of \( \text{order} \)) in the array \( \text{v} \).
    
    Constraints:
    
    \[
    \begin{align*}
    \text{if } \text{order} = \text{Nag\_ColMajor}, \ & pdv \geq \max(1,n); \\
    \text{if } \text{order} = \text{Nag\_RowMajor}, \ & pdv \geq \max(1,m).
    \end{align*}
    \]

11: \( \text{fail} \) – NagError *
    \textit{Output}
    
    The NAG error parameter (see the Essential Introduction).

### 6 Error Indicators and Warnings

**NE\_INT**

On entry, \( n = \langle \text{value} \rangle \).

Constraint: \( n \geq 0 \).

On entry, \( m = \langle \text{value} \rangle \).

Constraint: \( m \geq 0 \).

On entry, \( pdv = \langle \text{value} \rangle \).

Constraint: \( pdv > 0 \).
NE_INT_2

On entry, \( pdv = \langle value \rangle \), \( n = \langle value \rangle \).
Constraint: \( pdv \geq \max(1, n) \).

On entry, \( pdv = \langle value \rangle \), \( m = \langle value \rangle \).
Constraint: \( pdv \geq \max(1, m) \).

NE_INT_3

On entry, \( n = \langle value \rangle \), \(ilo = \langle value \rangle \), \(ihi = \langle value \rangle \).
Constraint: if \( n > 0 \), \( 1 \leq ilo \leq ihi \leq n \);
if \( n = 0 \), \( ilo = 1 \) and \( ihi = 0 \).

NE_ALLOC_FAIL

Memory allocation failed.

NE_BAD_PARAM

On entry, parameter \( \langle value \rangle \) had an illegal value.

NE_INTERNAL_ERROR

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please consult NAG for assistance.

7 Accuracy

The errors are negligible.

8 Further Comments

The total number of real floating-point operations is approximately proportional to \( nm \).

The real analogue of this function is nag_dgebak (f08njc).

9 Example

See Section 9 of the document for nag_zgebal (f08nvc).