NAG C Library Function Document

nag_dgebak (f08nqc)

1 Purpose

nag_dgebak (f08nqc) transforms eigenvectors of a balanced matrix to those of the original real nonsymmetric matrix.

2 Specification

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

3 Description

nag_dgebak (f08nqc) is intended to be used after a real nonsymmetric matrix $A$ has been balanced by nag_dgebal (f08nhc), and eigenvectors of the balanced matrix $A^{00}$ have subsequently been computed.

For a description of balancing, see the document for nag_dgebal (f08nhc). The balanced matrix $A^{00}$ is obtained as $A^{00} = DPAP^T D^{-1}$, 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 $A^{00}$, $P^T D^{-1} x$ is a right eigenvector of $A$;
- if $y$ is a left eigenvector of $A^{00}$, $P^T D y$ is a left eigenvector of $A$.

4 References

None.

5 Parameters

1: order – Nag_OrderType

   Input
   
   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

   Input
   
   On entry: this must be the same parameter job as supplied to nag_dgebal (f08nhc).
   
   Constraint: job = Nag_DoNothing, Nag_Permute, Nag_Scale or Nag_DoBoth.

3: side – Nag_SideType

   Input
   
   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}

\textit{On entry:} \( n \), the number of rows of the matrix of eigenvectors.

\textit{Constraint:} \( n \geq 0 \).

5: \( \text{i}0 \) – Integer

\textit{Input}

6: \( \text{i}1 \) – Integer

\textit{Input}

\textit{On entry:} the values \( i_{\text{lo}} \) and \( i_{\text{hi}} \), as returned by \texttt{nag_dgebal (f08nhc)}.

\textit{Constraints:}

\begin{align*}
&\text{if} \ n > 0, \ 1 \leq i_{\text{lo}} \leq i_{\text{hi}} \leq n; \\
&\text{if} \ n = 0, \ i_{\text{lo}} = 1 \ \text{and} \ i_{\text{hi}} = 0.
\end{align*}

7: \( \text{scale}[\text{dim}] \) – const double

\textit{Input}

\textit{Note:} the dimension, \( \text{dim} \), of the array \( \text{scale} \) must be at least \( \max(1, n) \).

\textit{On entry:} details of the permutations and/or the scaling factors used to balance the original real nonsymmetric matrix, as returned by \texttt{nag_dgebal (f08nhc)}.

8: \( m \) – Integer

\textit{Input}

\textit{On entry:} \( m \), the number of columns of the matrix of eigenvectors.

\textit{Constraint:} \( m \geq 0 \).

9: \( \text{v}[\text{dim}] \) – double

\textit{Input/Output}

\textit{Note:} the dimension, \( \text{dim} \), of the array \( \text{v} \) must be at least \( \max(1, \text{pdv} \times m) \) when \texttt{order = Nag_ColMajor} and at least \( \max(1, \text{pdv} \times n) \) when \texttt{order = Nag_RowMajor}.

If \texttt{order = Nag_ColMajor}, the \((i, j)\)th element of the matrix \( V \) is stored in \( \text{v}[(j - 1) \times \text{pdv} + i - 1] \) and if \texttt{order = Nag_RowMajor}, the \((i, j)\)th element of the matrix \( V \) is stored in \( \text{v}[(i - 1) \times \text{pdv} + j - 1] \).

\textit{On entry:} the matrix of left or right eigenvectors to be transformed.

\textit{On exit:} the transformed eigenvectors.

10: \( \text{pdv} \) – Integer

\textit{Input}

\textit{On entry:} the stride separating matrix row or column elements (depending on the value of \texttt{order}) in the array \( \text{v} \).

\textit{Constraints:}

\begin{align*}
&\text{if} \ \texttt{order = Nag_ColMajor}, \ \text{pdv} \geq \max(1, n); \\
&\text{if} \ \texttt{order = Nag_RowMajor}, \ \text{pdv} \geq \max(1, m).
\end{align*}

11: \( \text{fail} \) – \texttt{NagError *}

\textit{Output}

The \texttt{NagError} * parameter (see the Essential Introduction).

6 \ Error Indicators and Warnings

\textbf{NE_INT}

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

\textit{Constraint:} \( n \geq 0 \).

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

\textit{Constraint:} \( m \geq 0 \).

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

\textit{Constraint:} \( \text{pdv} > 0 \).
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 floating-point operations is approximately proportional to \( nm \).
The complex analogue of this function is \text{nag_zgebak (f08nwc)}.

9 Example
See Section 9 of the document for \text{nag_dgebal (f08nhc)}.