NAG C Library Function Document

nag_ztrevc (f08qxc)

1 Purpose

nag_ztrevc (f08qxc) computes selected left and/or right eigenvectors of a complex upper triangular matrix.

2 Specification

```c
void nag_ztrevc (Nag_OrderType order, Nag_SideType side, Nag_HowManyType how_many, 
    const Boolean select[], Integer n, Complex t[], Integer pdt, Complex vl[], 
    Integer pdvl, Complex vr[], Integer pdvr, Integer mm, Integer *m, 
    NagError *fail)
```

3 Description

nag_ztrevc (f08qxc) computes left and/or right eigenvectors of a complex upper triangular matrix $T$. Such a matrix arises from the Schur factorization of a complex general matrix, as computed by nag_zhseqr (f08psc), for example.

The right eigenvector $x$, and the left eigenvector $y$, corresponding to an eigenvalue $\lambda$, are defined by:

$$T x = \lambda x \quad \text{and} \quad y^H T = \lambda y^H \quad \text{(or} \quad T^H y = \bar{\lambda} y)\).$$

The function can compute the eigenvectors corresponding to selected eigenvalues, or it can compute all the eigenvectors. In the latter case the eigenvectors may optionally be pre-multiplied by an input matrix $Q$. Normally $Q$ is a unitary matrix from the Schur factorization of a matrix $A$ as $A = QTQ^H$; if $x$ is a (left or right) eigenvector of $T$, then $Qx$ is an eigenvector of $A$.

The eigenvectors are computed by forward or backward substitution. They are scaled so that $\max(\{|\text{Re}(x_i)| + |\text{Im}(x_i)|\}) = 1$.

4 References


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:  
   **side** – Nag_SideType

   *Input*

   **On entry:** indicates whether left and/or right eigenvectors are to be computed as follows:

   - if `side = Nag_RightSide`, only right eigenvectors are computed;
   - if `side = Nag_LeftSide`, only left eigenvectors are computed;
   - if `side = Nag_BothSides`, both left and right eigenvectors are computed.

   **Constraint:** `side = Nag_RightSide, Nag_LeftSide` or `Nag_BothSides`.  

[NP3645/7]
3:  how_many – Nag_HowManyType  
   Input

   On entry: indicates how many eigenvectors are to be computed as follows:
   
   if how_many = Nag_ComputeAll, all eigenvectors (as specified by side) are computed;
   if how_many = Nag_BackTransform, all eigenvectors (as specified by side) are computed
   and then pre-multiplied by the matrix Q (which is overwritten);
   if how_many = Nag_ComputeSelected, selected eigenvectors (as specified by side and
   select) are computed.

   Constraint: how_many = Nag_ComputeAll, Nag_BackTransform or Nag_ComputeSelected.

4:  select[dim] – const Boolean  
   Input

   Note: the dimension, dim, of the array select must be at least max(1,n) when
   how_many = Nag_ComputeSelected and at least 1 otherwise.

   On entry: select specifies which eigenvectors are to be computed if
   how_many = Nag_ComputeSelected. To obtain the eigenvector corresponding to the eigenvalue
   \( \lambda_j \), select[j] must be set TRUE.

   select is not referenced if how_many = Nag_ComputeAll or Nag_BackTransform.

5:  n – Integer  
   Input

   On entry: n, the order of the matrix \( T \).

   Constraint: n \( \geq 0 \).

6:  t[dim] – Complex  
   Input/Output

   Note: the dimension, dim, of the array t must be at least max(1,pdt \( \times n \)).

   If order = Nag_ColMajor, the \((i,j)\)th element of the matrix \( T \) is stored in
   \( t[(j-1) \times pdt + i - 1] \) and
   if order = Nag_RowMajor, the \((i,j)\)th element of the matrix \( T \) is stored in
   \( t[(i-1) \times pdt + j - 1] \).

   On entry: the \( n \) by \( n \) upper triangular matrix \( T \), as returned by nag_zhseqr (f08psc).

   On exit: \( t \) is used as internal workspace prior to being restored and hence is unchanged.

7:  pdt – Integer  
   Input

   On entry: the stride separating matrix row or column elements (depending on the value of order) in
   the array \( t \).

   Constraint: pdt \( \geq \) max(1,n).

8:  vl[dim] – Complex  
   Input/Output

   Note: the dimension, dim, of the array \( vl \) must be at least
   
   \[ \text{max}(1, pdvl \times \text{mm}) \] when
   side = Nag_LeftSide or Nag_BothSides and
   order = Nag_ColMajor;
   \[ \text{max}(1, pdvl \times n) \] when
   side = Nag_LeftSide or Nag_BothSides and
   order = Nag_RowMajor;
   \[ 1 \] when side = Nag_RightSide.

   If order = Nag_ColMajor, the \((i,j)\)th element of the matrix is stored in
   \( vl[(j-1) \times pdvl + i - 1] \) and
   if order = Nag_RowMajor, the \((i,j)\)th element of the matrix is stored in
   \( vl[(i-1) \times pdvl + j - 1] \).

   On entry: if how_many = Nag_BackTransform and side = Nag_LeftSide or Nag_BothSides, \( vl \)
   must contain an \( n \) by \( n \) matrix \( Q \) (usually the matrix of Schur vectors returned by
   nag_zhseqr (f08psc)). If how_many = Nag_ComputeAll or Nag_ComputeSelected, \( vl \) need not be set.

   On exit: if side = Nag_LeftSide or Nag_BothSides, \( vl \) contains the computed left eigenvectors (as
   specified by how_many and select). The eigenvectors are stored consecutively in the rows or
   columns (depending on the value of order) of the array, in the same order as their eigenvalues.
vl is not referenced if \texttt{side} = \texttt{Nag\_RightSide}.

9: \begin{align*}
\texttt{pdvl} & \quad \text{Integer} \quad \textit{Input} \\
\textit{On entry}: \text{the stride separating matrix row or column elements (depending on the value of } \texttt{order} \text{) in the array } \texttt{vl}. \\
\textit{Constraints}: & \\
\text{if } \texttt{order} = \texttt{Nag\_ColMajor}, & \\
& \text{if } \texttt{side} = \texttt{Nag\_LeftSide} \text{ or } \texttt{Nag\_BothSides}, \texttt{pdvl} \geq \text{max}(1, n); \\
& \text{if } \texttt{side} = \texttt{Nag\_RightSide}, \texttt{pdvl} \geq 1; \\
\text{if } \texttt{order} = \texttt{Nag\_RowMajor}, & \\
& \text{if } \texttt{side} = \texttt{Nag\_LeftSide} \text{ or } \texttt{Nag\_BothSides}, \texttt{pdvl} \geq \text{max}(1, \texttt{mm}); \\
& \text{if } \texttt{side} = \texttt{Nag\_RightSide}, \texttt{pdvl} \geq 1.
\end{align*}

10: \begin{align*}
\text{\texttt{vr}[dim]} & \quad \text{Complex} \quad \textit{Input/Output} \\
\textit{Note}: \text{the dimension, } \texttt{dim}, \text{ of the array } \texttt{vr} \text{ must be at least} & \\
\text{max}(1, \texttt{pdvr} \times \texttt{mm}) & \text{ when } \texttt{side} = \texttt{Nag\_RightSide} \text{ or } \texttt{Nag\_BothSides} \text{ and } \texttt{order} = \texttt{Nag\_ColMajor}; \\
\text{max}(1, \texttt{pdvr} \times n) & \text{ when } \texttt{side} = \texttt{Nag\_RightSide} \text{ or } \texttt{Nag\_BothSides} \text{ and } \texttt{order} = \texttt{Nag\_RowMajor}; \\
1 & \text{ when } \texttt{side} = \texttt{Nag\_LeftSide}. \\
\text{If } \texttt{order} = \texttt{Nag\_ColMajor}, \text{ the } (i, j)\text{th element of the matrix is stored in } \texttt{vr}[j - 1 \times \texttt{pdvr} + i - 1] \text{ and} & \\
\text{if } \texttt{order} = \texttt{Nag\_RowMajor}, \text{ the } (i, j)\text{th element of the matrix is stored in } \texttt{vr}[i - 1 \times \texttt{pdvr} + j - 1]. \\
\textit{On entry}: \text{if } \texttt{how\_many} = \texttt{Nag\_BackTransform} \text{ and } \texttt{side} = \texttt{Nag\_RightSide} \text{ or } \texttt{Nag\_BothSides}, \texttt{vr} \text{ must contain an } n \text{ by } n \text{ matrix } Q \text{ (usually the matrix of Schur vectors returned by } \texttt{nag\_zhseqr} (f08psc)). \text{ If } \texttt{how\_many} = \texttt{Nag\_ComputeAll} \text{ or } \texttt{Nag\_ComputeSelected}, \texttt{vr} \text{ need not be set.} \\
\textit{On exit}: \text{if } \texttt{side} = \texttt{Nag\_RightSide} \text{ or } \texttt{Nag\_BothSides}, \texttt{vr} \text{ contains the computed right eigenvectors (as specified by } \texttt{how\_many} \text{ and } \texttt{select}). \text{ The eigenvectors are stored consecutively in the rows or columns (depending on the value of } \texttt{order}) \text{ of the array, in the same order as their eigenvalues.} \\
\texttt{vr} \text{ is not referenced if } \texttt{side} = \texttt{Nag\_LeftSide}.
\end{align*}

11: \begin{align*}
\texttt{pdvr} & \quad \text{Integer} \quad \textit{Input} \\
\textit{On entry}: \text{the stride separating matrix row or column elements (depending on the value of } \texttt{order} \text{) in the array } \texttt{vr}. \\
\textit{Constraints}: & \\
\text{if } \texttt{order} = \texttt{Nag\_ColMajor}, & \\
& \text{if } \texttt{side} = \texttt{Nag\_RightSide} \text{ or } \texttt{Nag\_BothSides}, \texttt{pdvr} \geq \text{max}(1, n); \\
& \text{if } \texttt{side} = \texttt{Nag\_LeftSide}, \texttt{pdvr} \geq 1; \\
\text{if } \texttt{order} = \texttt{Nag\_RowMajor}, & \\
& \text{if } \texttt{side} = \texttt{Nag\_RightSide} \text{ or } \texttt{Nag\_BothSides}, \texttt{pdvr} \geq \text{max}(1, \texttt{mm}); \\
& \text{if } \texttt{side} = \texttt{Nag\_LeftSide}, \texttt{pdvr} \geq 1.
\end{align*}

12: \begin{align*}
\texttt{mm} & \quad \text{Integer} \quad \textit{Input} \\
\textit{On entry}: \text{the number of rows or columns (depending on the value of } \texttt{order} \text{) in the arrays } \texttt{vl} \text{ and/or } \texttt{vr}. \text{ The precise number of rows or columns required, } \texttt{required\_rowcol}, \text{ is } n \text{ if } \texttt{how\_many} = \texttt{Nag\_ComputeAll} \text{ or } \texttt{Nag\_BackTransform}; \text{ if } \texttt{how\_many} = \texttt{Nag\_ComputeSelected}, \texttt{required\_rowcol} \text{ is the number of selected eigenvectors (see } \texttt{select}), \text{ in which case } 0 \leq \texttt{required\_rowcol} \leq n. \\
\textit{Constraint}: \texttt{mm} \geq \texttt{required\_rowcol}.
\end{align*}
13: \( m \) – Integer *  
\textit{Output}

\textit{On exit: required\_rowcol}, the number of selected eigenvectors. If \texttt{how\_many} = \texttt{Nag\_ComputeAll} or \texttt{Nag\_BackTransform}, \( m \) is set to \( n \).

14: \( \text{fail} \) – NagError *  
\textit{Output}

The NAG error parameter (see the Essential Introduction).

6 \ Error Indicators and Warnings

\textbf{NE\_INT}

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

On entry, \( mm = \langle \text{value} \rangle \).
\text{Constraint:} \( mm \geq \text{required\_rowcol} \), where \text{required\_rowcol} is the number of selected eigenvectors.

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

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

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

\textbf{NE\_INT\_2}

On entry, \( \text{pdt} = \langle \text{value} \rangle, n = \langle \text{value} \rangle \).
\text{Constraint:} \( \text{pdt} \geq \text{max}(1, n) \).

\textbf{NE\_ENUM\_INT\_2}

On entry, \( \text{side} = \langle \text{value} \rangle, n = \langle \text{value} \rangle, \text{pdvl} = \langle \text{value} \rangle \).
\text{Constraint:} if \( \text{side} = \texttt{Nag\_LeftSide} \) or \( \texttt{Nag\_BothSides} \), \( \text{pdvl} \geq \text{max}(1, n) \); if \( \text{side} = \texttt{Nag\_RightSide} \), \( \text{pdvl} \geq 1 \).

On entry, \( \text{side} = \langle \text{value} \rangle, n = \langle \text{value} \rangle, \text{pdvr} = \langle \text{value} \rangle \).
\text{Constraint:} if \( \text{side} = \texttt{Nag\_RightSide} \) or \( \texttt{Nag\_BothSides} \), \( \text{pdvr} \geq \text{max}(1, n) \); if \( \text{side} = \texttt{Nag\_LeftSide} \), \( \text{pdvr} \geq 1 \).

On entry, \( \text{side} = \langle \text{value} \rangle, mm = \langle \text{value} \rangle, \text{pdvl} = \langle \text{value} \rangle \).
\text{Constraint:} if \( \text{side} = \texttt{Nag\_LeftSide} \) or \( \texttt{Nag\_BothSides} \), \( \text{pdvl} \geq \text{max}(1, mm) \); if \( \text{side} = \texttt{Nag\_RightSide} \), \( \text{pdvl} \geq 1 \).

On entry, \( \text{side} = \langle \text{value} \rangle, mm = \langle \text{value} \rangle, \text{pdvr} = \langle \text{value} \rangle \).
\text{Constraint:} if \( \text{side} = \texttt{Nag\_RightSide} \) or \( \texttt{Nag\_BothSides} \), \( \text{pdvr} \geq \text{max}(1, mm) \); if \( \text{side} = \texttt{Nag\_LeftSide} \), \( \text{pdvr} \geq 1 \).

\textbf{NE\_ALLOC\_FAIL}

Memory allocation failed.

\textbf{NE\_BAD\_PARAM}

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

\textbf{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

If \( x_i \) is an exact right eigenvector, and \( \tilde{x}_i \) is the corresponding computed eigenvector, then the angle \( \theta(\tilde{x}_i, x_i) \) between them is bounded as follows:

\[
\theta(\tilde{x}_i, x_i) \leq \frac{c(n)\varepsilon\|T\|_2}{\text{sep}_i}
\]

where \( \text{sep}_i \) is the reciprocal condition number of \( x_i \).

The condition number \( \text{sep}_i \) may be computed by calling nag_ztrsna (f08qyc).

8 Further Comments

The real analogue of this function is nag_dtrevc (f08qkc).

9 Example

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