NAG C Library Function Document

nag_zhsein (f08pxc)

1 Purpose
nag_zhsein (f08pxc) computes selected left and/or right eigenvectors of a complex upper Hessenberg
matrix corresponding to specified eigenvalues, by inverse iteration.

2 Specification
void nag_zhsein (Nag_OrderType order, Nag_SideType side,
Nag_EigValsSourceType eig_source, Nag_InitVeenumtype initv,
const Boolean select[], Integer n, const Complex h[], Integer pdh, Complex w[],
Complex vl[], Integer pdvl, Complex vr[], Integer pdvr, Integer mm,
Integer *m, Integer ifail[], Integer ifailr[], NagError *fail)

3 Description
nag_zhsein (f08pxc) computes left and/or right eigenvectors of a complex upper Hessenberg matrix $H$,
corresponding to selected eigenvalues.

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

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

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

If $H$ has been formed by reduction of a complex general matrix $A$ to upper Hessenberg form, then the
eigenvectors of $H$ may be transformed to eigenvectors of $A$ by a call to nag_zunmhr (f08nuc).

4 References
Baltimore

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.
eig_source – Nag_EigValsSourceType  
*Input*

*On entry:* indicates whether the eigenvalues of $H$ (stored in $w$) were found using nag_zhseqr (f08psc) as follows:

- If $\text{eig\_source} = \text{Nag\_HSEQRSource}$, then the eigenvalues of $H$ were found using nag_zhseqr (f08psc); thus if $H$ has any zero sub-diagonal elements (and so is block triangular), then the $j$th eigenvalue can be assumed to be an eigenvalue of the block containing the $j$th row/column. This property allows the function to perform inverse iteration on just one diagonal block;

- If $\text{eig\_source} = \text{Nag\_NotKnown}$, then no such assumption is made and the function performs inverse iteration using the whole matrix.

*Constraint:* $\text{eig\_source} = \text{Nag\_HSEQRSource}$ or $\text{Nag\_NotKnown}$.

initv – Nag_InitVeenumtype  
*Input*

*On entry:* indicates whether the user is supplying initial estimates for the selected eigenvectors as follows:

- If $\text{initv} = \text{Nag\_NoVec}$, no initial estimates are supplied;
- If $\text{initv} = \text{Nag\_UserVec}$, initial estimates are supplied in $vl$ and/or $vr$.

*Constraint:* $\text{initv} = \text{Nag\_NoVec}$ or $\text{Nag\_UserVec}$.

select[|dim|] – const Boolean  
*Input*

*Note:* the dimension, $dim$, of the array select must be at least $\max(1, n)$.

*On entry:* select specifies which eigenvectors are to be computed. To select the eigenvector corresponding to the eigenvalue $w[j]$, select[$j$] must be set to TRUE.

n – Integer  
*Input*

*On entry:* $n$, the order of the matrix $H$.

*Constraint:* $n \geq 0$.

h[|dim|] – const Complex  
*Input*

*Note:* the dimension, $dim$, of the array $h$ must be at least $\max(1, pdh \times n)$.

If $\text{order} = \text{Nag\_ColMajor}$, the $(i, j)$th element of the matrix $H$ is stored in $h[(j - 1) \times \text{pdh} + i - 1]$ and if $\text{order} = \text{Nag\_RowMajor}$, the $(i, j)$th element of the matrix $H$ is stored in $h[(i - 1) \times \text{pdh} + j - 1]$.

*On entry:* the $n$ by $n$ upper Hessenberg matrix $H$.

pdh – Integer  
*Input*

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

*Constraint:* $\text{pdh} \geq \max(1, n)$.

w[|dim|] – Complex  
*Input/Output*

*Note:* the dimension, $dim$, of the array $w$ must be at least $\max(1, n)$.

*On entry:* the eigenvalues of the matrix $H$. If $\text{eig\_source} = \text{Nag\_HSEQRSource}$, the array must be exactly as returned by nag_zhseqr (f08psc).

*On exit:* the real parts of some elements of $w$ may be modified, as close eigenvalues are perturbed slightly in searching for independent eigenvectors.
10: \textbf{vl[dim]} – Complex \hspace{1cm} \textit{Input/Output}

\textbf{Note:} the dimension, \textit{dim}, of the array \textit{vl} must be at least
\[
\max(1, \text{pdvl} \times \text{mm}) \quad \text{when} \quad \textit{side} = \text{Nag\_LeftSide} \quad \text{or} \quad \text{Nag\_BothSides} \quad \text{and} \quad \textit{order} = \text{Nag\_ColMajor};
\]
\[
\max(1, \text{pdvl} \times n) \quad \text{when} \quad \textit{side} = \text{Nag\_LeftSide} \quad \text{or} \quad \text{Nag\_BothSides} \quad \text{and} \quad \textit{order} = \text{Nag\_RowMajor};
\]
1 \quad \text{when} \quad \textit{side} = \text{Nag\_RightSide}.

If \textit{order} = \text{Nag\_ColMajor}, the \((i, j)\)th element of the matrix is stored in \textit{vl}[(\(j - 1\) \times \text{pdvl} + i - 1)] and if \textit{order} = \text{Nag\_RowMajor}, the \((i, j)\)th element of the matrix is stored in \textit{vl}[(i - 1) \times \text{pdvl} + j - 1].

\textbf{On entry:} if \textit{initv} = \text{Nag\_UserVec} and \textit{side} = \text{Nag\_LeftSide} or \text{Nag\_BothSides}, \textit{vl} must contain starting vectors for inverse iteration for the left eigenvectors. Each starting vector must be stored in the same row or column as will be used to store the corresponding eigenvector (see below). If \textit{initv} = \text{Nag\_NoVec}, \textit{vl} need not be set.

\textbf{On exit:} if \textit{side} = \text{Nag\_LeftSide} or \text{Nag\_BothSides}, \textit{vl} contains the computed left eigenvectors (as specified by \textit{select}). The eigenvectors are stored consecutively in the rows or columns of the array (depending on the value of \textit{order}), in the same order as their eigenvalues.

\textit{vl} is not referenced if \textit{side} = \text{Nag\_RightSide}.

11: \textbf{pdvl} – Integer \hspace{1cm} \textit{Input}

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

\textbf{Constraints:}

if \textit{order} = \text{Nag\_ColMajor},
\[
\text{if} \quad \textit{side} = \text{Nag\_LeftSide} \quad \text{or} \quad \text{Nag\_BothSides}, \quad \text{pdvl} \geq \max(1, n);
\]
\[
\text{if} \quad \textit{side} = \text{Nag\_RightSide}, \quad \text{pdvl} \geq 1;
\]
if \textit{order} = \text{Nag\_RowMajor},
\[
\text{if} \quad \textit{side} = \text{Nag\_LeftSide} \quad \text{or} \quad \text{Nag\_BothSides}, \quad \text{pdvl} \geq \max(1, \text{mm});
\]
\[
\text{if} \quad \textit{side} = \text{Nag\_RightSide}, \quad \text{pdvl} \geq 1.
\]

12: \textbf{vr[dim]} – Complex \hspace{1cm} \textit{Input/Output}

\textbf{Note:} the dimension, \textit{dim}, of the array \textit{vr} must be at least
\[
\max(1, \text{pdvr} \times \text{mm}) \quad \text{when} \quad \textit{side} = \text{Nag\_RightSide} \quad \text{or} \quad \text{Nag\_BothSides} \quad \text{and} \quad \textit{order} = \text{Nag\_ColMajor};
\]
\[
\max(1, \text{pdvr} \times n) \quad \text{when} \quad \textit{side} = \text{Nag\_RightSide} \quad \text{or} \quad \text{Nag\_BothSides} \quad \text{and} \quad \textit{order} = \text{Nag\_RowMajor};
\]
1 \quad \text{when} \quad \textit{side} = \text{Nag\_LeftSide}.

If \textit{order} = \text{Nag\_ColMajor}, the \((i, j)\)th element of the matrix is stored in \textit{vr}[(\(j - 1\) \times \text{pdvr} + i - 1)] and if \textit{order} = \text{Nag\_RowMajor}, the \((i, j)\)th element of the matrix is stored in \textit{vr}[(i - 1) \times \text{pdvr} + j - 1].

\textbf{On entry:} if \textit{initv} = \text{Nag\_UserVec} and \textit{side} = \text{Nag\_RightSide} or \text{Nag\_BothSides}, \textit{vr} must contain starting vectors for inverse iteration for the right eigenvectors. Each starting vector must be stored in the same row or column as will be used to store the corresponding eigenvector (see below). If \textit{initv} = \text{Nag\_NoVec}, \textit{vr} need not be set.

\textbf{On exit:} if \textit{side} = \text{Nag\_RightSide} or \text{Nag\_BothSides}, \textit{vr} contains the computed right eigenvectors (as specified by \textit{select}). The eigenvectors are stored consecutively in the rows or columns of the array (depending on the value of \textit{order}), in the same order as their eigenvalues.

\textit{vr} is not referenced if \textit{side} = \text{Nag\_LeftSide}.

13: \textbf{pdvr} – Integer \hspace{1cm} \textit{Input}

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

[NP3645/7] f08pxc.3
Constraints:

if order = Nag_ColMajor,
  if side = Nag_RightSide or Nag_BothSides, pdvr \geq \max(1, n);
  if side = Nag_LeftSide, pdvr \geq 1;

if order = Nag_RowMajor,
  if side = Nag_RightSide or Nag_BothSides, pdvr \geq \max(1, mm);
  if side = Nag_LeftSide, pdvr \geq 1.

14: \text{mm} \quad \text{– Integer} \quad \text{Input}

On entry: the number of columns in the arrays vl and/or vr if order = Nag_ColMajor or the number of rows in the arrays if order = Nag_RowMajor. The actual number of rows or columns required, \text{required_rowcol}, is obtained by counting 1 for each selected real eigenvector and 2 for each selected complex eigenvector (see \text{select}); 0 \leq \text{required_rowcol} \leq n.

Constraint: \text{mm} \geq \text{required_rowcol}.

15: \text{m} \quad \text{– Integer} \quad \text{Output}

On exit: \text{required_rowcol}, the number of selected eigenvectors.

16: \text{ifail}[\dim] \quad \text{– Integer} \quad \text{Output}

Note: the dimension, \text{dim}, of the array \text{ifail} must be at least \max(1, \text{mm}) when \text{side} = Nag_LeftSide or Nag_BothSides and at least 1 when \text{side} = Nag_RightSide.

On exit: if \text{side} = Nag_LeftSide or Nag_BothSides, then ifail[i] = 0 if the selected left eigenvector converged and ifail[i] = j \geq 0 if the eigenvector stored in the i\text{th} row or column of vl (corresponding to the j\text{th} eigenvalue) failed to converge.

ifail is not referenced if \text{side} = Nag_RightSide.

17: \text{ifailr}[\dim] \quad \text{– Integer} \quad \text{Output}

Note: the dimension, \text{dim}, of the array \text{ifailr} must be at least \max(1, \text{mm}) when \text{side} = Nag_RightSide or Nag_BothSides and at least 1 when \text{side} = Nag_LeftSide.

On exit: if \text{side} = Nag_RightSide or Nag_BothSides, then ifailr[i] = 0 if the selected right eigenvector converged and ifailr[i] = j \geq 0 if the eigenvector stored in the i\text{th} column of vr (corresponding to the j\text{th} eigenvalue) failed to converge.

ifailr is not referenced if \text{side} = Nag_LeftSide.

18: \text{fail} \quad \text{– NagError} \quad \text{Output}

The NAG error parameter (see the Essential Introduction).

6 Error Indicators and Warnings

NE_INT

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

Constraint: \text{n} \geq 0.

On entry, \text{mm} = \langle value\rangle.

Constraint: \text{mm} \geq \text{required_rowcol}, where \text{required_rowcol} is the number of selected eigenvectors.

On entry, \text{pdh} = \langle value\rangle.

Constraint: \text{pdh} > 0.

On entry, \text{pdvl} = \langle value\rangle.

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

NE_INT_2
On entry, pdh = (value), n = (value).
Constraint: pdh ≥ max(1, n).

NE_ENUM_INT_2
On entry, side = (value), n = (value), pdvl = (value).
Constraint: if side = Nag_LeftSide or Nag_BothSides, pdvl ≥ max(1, n);
if side = Nag_RightSide, pdvl ≥ n.
On entry, side = (value), mm = (value), pdvl = (value).
Constraint: if side = Nag_LeftSide or Nag_BothSides, pdvl ≥ max(1, mm);
if side = Nag_RightSide, pdvl ≥ mm.
On entry, side = (value), mm = (value), pdvr = (value).
Constraint: if side = Nag_LeftSide or Nag_BothSides, pdvr ≥ max(1, mm);
if side = Nag_RightSide, pdvr ≥ n.

NE_CONVERGENCE
(value) eigenvectors (as indicated by arguments ifaill and/or ifailr) failed to converge. The corresponding columns of vl and/or vr contain no useful information.

NE_ALLOC_FAIL
Memory allocation failed.

NE_BAD_PARAM
On entry, parameter (value) 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
Each computed right eigenvector $x_i$ is the exact eigenvector of a nearby matrix $A + E_i$, such that $||E_i|| = O(\epsilon)||A||$. Hence the residual is small:

$$||Ax_i - \lambda_i x_i|| = O(\epsilon)||A||.$$ 

However, eigenvectors corresponding to close or coincident eigenvalues may not accurately span the relevant subspaces.

Similar remarks apply to computed left eigenvectors.

8 Further Comments
The real analogue of this function is nag_dhsein (f08pke).

9 Example
See Section 9 of the document for nag_zunmhr (f08nuc).