NAG C Library Function Document

nag_zstein (f08jxc)

1 Purpose

nag_zstein (f08jxc) computes the eigenvectors of a real symmetric tridiagonal matrix corresponding to specified eigenvalues, by inverse iteration, storing the eigenvectors in a complex array.

2 Specification

```c
void nag_zstein (Nag_OrderType order, Integer n, const double d[], const double e[],
               Integer m, const double w[], const Integer iblock[], const Integer isplit[],
               Complex z[], Integer pdz, Integer ifailv[], NagError *fail)
```

3 Description

nag_zstein (f08jxc) computes the eigenvectors of a real symmetric tridiagonal matrix $T$ corresponding to specified eigenvalues, by inverse iteration (see Jessup and Ipsen (1992)). It is designed to be used in particular after the specified eigenvalues have been computed by nag_dstebz (f08jjc) with `order = Nag_ByBlock`, but may also be used when the eigenvalues have been computed by other f08 or f02 functions.

The eigenvectors of $T$ are real, but are stored by this function in a complex array. If $T$ has been formed by reduction of a full complex Hermitian matrix $A$ to tridiagonal form, then eigenvectors of $T$ may be transformed to (complex) eigenvectors of $A$, by a call to nag_zunmtr (f08fuc) or nag_zupmtr (f08guc).

nag_dstebz (f08jjc) determines whether the matrix $T$ splits into block diagonal form:

$$T = \begin{pmatrix} T_1 & & \\ & T_2 & \\ & & \ddots \\ & & & T_p \end{pmatrix}$$

and passes details of the block structure to this function in the arrays `iblock` and `isplit`. This function can then take advantage of the block structure by performing inverse iteration on each block $T_i$ separately, which is more efficient than using the whole matrix.

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`. 
f08jxc

NAG C Library Manual

2: \( n \) – Integer

*Input*

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

*Constraint:* \( n \geq 0 \).

3: \( d[dim] \) – const double

*Input*

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

*On entry:* the diagonal elements of the tridiagonal matrix \( T \).

4: \( e[dim] \) – const double

*Input*

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

*On entry:* the off-diagonal elements of the tridiagonal matrix \( T \).

5: \( m \) – Integer

*Input*

*On entry:* \( m \), the number of eigenvectors to be returned.

*Constraint:* \( 0 \leq m \leq n \).

6: \( w[dim] \) – const double

*Input*

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

*On entry:* the eigenvalues of the tridiagonal matrix \( T \) stored in \( w[0] \) to \( w[m] \), as returned by nag_dstebz (f08jjc) with \( \text{rank} = \text{Nag}_\text{ByBlock} \). Eigenvalues associated with the first sub-matrix must be supplied first, in non-decreasing order; then those associated with the second sub-matrix, again in non-decreasing order; and so on.

*Constraint:* if \( \text{iblock}[i] = \text{iblock}[i + 1] \), \( w[i] \leq w[i + 1] \) for \( i = 0, 1, \ldots, m - 2 \).

7: \( \text{iblock[dim]} \) – const Integer

*Input*

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

*On entry:* the first \( m \) elements must contain the sub-matrix indices associated with the specified eigenvalues, as returned by nag_dstebz (f08jjc) with \( \text{order} = \text{Nag}_\text{ByBlock} \). If the eigenvalues were not computed by nag_dstebz (f08jjc) with \( \text{order} = \text{Nag}_\text{ByBlock} \), set \( \text{iblock}[i - 1] \) to 1 for \( i = 1, 2, \ldots, m \).

*Constraint:* \( \text{iblock}[i] \leq \text{iblock}[i + 1] \) for \( i = 0, 1, \ldots, m - 2 \).

8: \( \text{isplit[dim]} \) – const Integer

*Input*

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

*On entry:* the points at which \( T \) breaks up into sub-matrices, as returned by nag_dstebz (f08jjc) with \( \text{rank} = \text{Nag}_\text{ByBlock} \). If the eigenvalues were not computed by nag_dstebz (f08jjc) with \( \text{rank} = \text{Nag}_\text{ByBlock} \), set \( \text{isplit}[0] \) to \( n \).

9: \( z[dim] \) – Complex

*Output*

*Note:* the dimension, \( dim \), of the array \( z \) must be at least \( \max(1, \text{pdz} \times m) \) when \( \text{order} = \text{Nag}_\text{ColMajor} \) and at least \( \max(1, \text{pdz} \times n) \) when \( \text{order} = \text{Nag}_\text{RowMajor} \). If \( \text{order} = \text{Nag}_\text{ColMajor} \), the \((i, j)\)th element of the matrix \( Z \) is stored in \( z[(j - 1) \times \text{pdz} + i - 1] \) and if \( \text{order} = \text{Nag}_\text{RowMajor} \), the \((i, j)\)th element of the matrix \( Z \) is stored in \( z[(i - 1) \times \text{pdz} + j - 1] \).

*On exit:* the \( m \) eigenvectors, stored as columns of \( z \); the \( i \)th column corresponds to the \( i \)th specified eigenvalue, unless \( \text{fail} > 0 \) (in which case see Section 6).

10: \( \text{pdz} \) – Integer

*Input*

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

if order = Nag_ColMajor, pdz ≥ max(1, n);
if order = Nag_RowMajor, pdz ≥ max(1, m).

11: ifailv[dim] – Integer

Output

Note: the dimension, dim, of the array ifailv must be at least max(1, m).

On exit: if fail = i > 0, the first i elements of ifailv contain the indices of any eigenvectors which have failed to converge. The rest of the first m elements of ifailv are set to 0.

12: fail – NagError *

Output

The NAG error parameter (see the Essential Introduction).

6 Error Indicators and Warnings

NE_INT

On entry, n = ⟨value⟩.
Constraint: n ≥ 0.

On entry, pdz = ⟨value⟩.
Constraint: pdz > 0.

NE_INT_2

On entry, m = ⟨value⟩, n = ⟨value⟩.
Constraint: 0 ≤ m ≤ n.

On entry, pdz = ⟨value⟩, n = ⟨value⟩.
Constraint: pdz ≥ max(1, n).

On entry, pdz = ⟨value⟩, m = ⟨value⟩.
Constraint: pdz ≥ max(1, m).

NE_INT_ARRAY

On entry, iblock[i]w[i]iblock[i] = ⟨value⟩.
Constraint: if iblock[i] = iblock[i + 1], w[i] ≤ w[i + 1] for i = 0, . . . , m − 2.

On entry, iblock[i]w[i]iblock[i] = ⟨value⟩.
Constraint: iblock[i] ≤ iblock[i + 1] for i = 0, . . . , m − 2.

NE_CONVERGENCE

⟨value⟩ eigenvectors (as indicated by argument ifailv) each failed to converge in 5 iterations. The current iterate after 5 iterations is stored in the corresponding column of z.

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 eigenvector $z_i$ is the exact eigenvector of a nearby matrix $A + E_i$, such that $\|E_i\| = O(\epsilon)\|A\|$, where $\epsilon$ is the machine precision. Hence the residual is small:

$$\|Az_i - \lambda_i z_i\| = O(\epsilon)\|A\|.$$ 

However, a set of eigenvectors computed by this function may not be orthogonal to so high a degree of accuracy as those computed by nag_zsteqr (f08jsc).

8 Further Comments

The real analogue of this function is nag_dstein (f08jkc).

9 Example

See Section 9 of the document for nag_zunmtr (f08fuc).