NAG 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

```
#include <nag.h>
#include <nagf08.h>

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 **rank** $=$ Nag_ByBlock, but may also be used when the eigenvalues have been computed by other functions in Chapters f02 or f08.

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:

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

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

Golub G H and Van Loan C F (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

Jessup E and Ipsen I C F (1992) Improving the accuracy of inverse iteration *SIAM J. Sci. Statist. Comput.* **13** 550–572

5 Arguments

1: **order** – Nag_OrderType *Input*

On entry: the **order** argument 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 3.2.1.3 in the Essential Introduction for a more detailed explanation of the use of this argument.

Constraint: **order** = Nag_RowMajor or Nag_ColMajor.

2: **n** – Integer *Input*

On entry: n, the order of the matrix T.

Constraint: $\mathbf{n} \geq 0$.

3: $\mathbf{d}[dim]$ – const double *Input*

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

On entry: the diagonal elements of the tridiagonal matrix T.

4: $\mathbf{e}[dim]$ – const double *Input*

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

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: $0 \leq \mathbf{m} \leq \mathbf{n}$.

6: $\mathbf{w}[dim]$ – const double *Input*

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

On entry: the eigenvalues of the tridiagonal matrix T stored in $\mathbf{w}[0]$ to $\mathbf{w}[m-1]$, as returned by nag_dstebz (f08jjc) with **rank** = Nag_ByBlock. Eigenvalues associated with the first sub-matrix must be supplied first, in nondecreasing order; then those associated with the second sub-matrix, again in nondecreasing order; and so on.

Constraint: if $\mathbf{iblock}[i] = \mathbf{iblock}[i+1]$, $\mathbf{w}[i] \leq \mathbf{w}[i+1]$, for $i = 0, 1, \ldots, \mathbf{m} - 2$.

7: $\mathbf{iblock}[dim]$ – const Integer *Input*

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

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

Constraint: $\mathbf{iblock}[i] \leq \mathbf{iblock}[i+1]$, for $i = 0, 1, \ldots, \mathbf{m} - 2$.

8: $\mathbf{isplit}[dim]$ – const Integer *Input*

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

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

9: $\mathbf{z}[dim]$ – Complex *Output*

Note: the dimension, *dim*, of the array **z** must be at least

$\max(1, \mathbf{pdz} \times \mathbf{m})$ when **order** = Nag_ColMajor;
$\max(1, \mathbf{n} \times \mathbf{pdz})$ when **order** = Nag_RowMajor.

The (i,j)th element of the matrix Z is stored in

$\mathbf{z}[(j-1) \times \mathbf{pdz} + i - 1]$ when $\mathbf{order} = \text{Nag\_ColMajor}$;
$\mathbf{z}[(i-1) \times \mathbf{pdz} + j - 1]$ when $\mathbf{order} = \text{Nag\_RowMajor}$.

On exit: the m eigenvectors, stored as columns of Z; the ith column corresponds to the ith specified eigenvalue, unless **fail.code** = NE_CONVERGENCE (in which case see Section 6).

10: **pdz** – Integer *Input*

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

Constraints:

if $\mathbf{order} = \text{Nag\_ColMajor}$, $\mathbf{pdz} \geq \max(1, \mathbf{n})$;
if $\mathbf{order} = \text{Nag\_RowMajor}$, $\mathbf{pdz} \geq \max(1, \mathbf{m})$.

11: **ifailv**[**m**] – Integer *Output*

On exit: if $\mathbf{fail.errnum} = i > 0$, 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 * *Input/Output*

The NAG error argument (see Section 3.6 in the Essential Introduction).

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.
See Section 3.2.1.2 in the Essential Introduction for further information.

NE_BAD_PARAM

On entry, argument ⟨*value*⟩ had an illegal value.

NE_CONSTRAINT

On entry, **m** = ⟨*value*⟩, $\mathbf{iblock}[i]\mathbf{iblock}[i+1]$ = ⟨*value*⟩ and $\mathbf{w}[i]\mathbf{w}[i+1]$ = ⟨*value*⟩.
Constraint: , for $i = 0, 1, \ldots, \mathbf{m} - 2$

NE_CONVERGENCE

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

NE_INT

On entry, **n** = ⟨*value*⟩.
Constraint: $\mathbf{n} \geq 0$.

On entry, **pdz** = ⟨*value*⟩.
Constraint: $\mathbf{pdz} > 0$.

NE_INT_2

On entry, **m** = ⟨*value*⟩ and **n** = ⟨*value*⟩.
Constraint: $0 \leq \mathbf{m} \leq \mathbf{n}$.

On entry, **pdz** = ⟨*value*⟩ and **m** = ⟨*value*⟩.
Constraint: $\mathbf{pdz} \geq \max(1, \mathbf{m})$.

On entry, **pdz** = ⟨*value*⟩ and **n** = ⟨*value*⟩.
Constraint: $\mathbf{pdz} \geq \max(1, \mathbf{n})$.

NE_INT_ARRAY

On entry, $\mathbf{m} = \langle value \rangle$ and $\mathbf{iblock}[i]\mathbf{iblock}[i+1] = \langle value \rangle$.
Constraint: $\mathbf{iblock}[i] \leq \mathbf{iblock}[i+1]$, for $i = 0, 1, \ldots, \mathbf{m} - 2$

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 contact NAG for assistance.

An unexpected error has been triggered by this function. Please contact NAG.
See Section 3.6.6 in the Essential Introduction for further information.

NE_NO_LICENCE

Your licence key may have expired or may not have been installed correctly.
See Section 3.6.5 in the Essential Introduction for further information.

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 ϵ 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 Parallelism and Performance

nag_zstein (f08jxc) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

nag_zstein (f08jxc) makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this function. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

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

10 Example

See Section 10 in nag_zunmtr (f08fuc).
