NAG Toolbox

nag_lapack_zstedc (f08jv)

1 Purpose

nag_lapack_zstedc (f08jv) computes all the eigenvalues and, optionally, all the eigenvectors of a real n by n symmetric tridiagonal matrix, or of a complex full or banded Hermitian matrix which has been reduced to tridiagonal form.

2 Syntax

```
[d, e, z, info] = nag_lapack_zstedc(compz, d, e, z, 'n', n)
[d, e, z, info] = f08jv(compz, d, e, z, 'n', n)
```

3 Description

nag_lapack_zstedc (f08jv) computes all the eigenvalues and, optionally, the eigenvectors of a real symmetric tridiagonal matrix T. That is, the function computes the spectral factorization of T given by

$$T = Z\Lambda Z^{\mathrm{T}},$$

where Λ is a diagonal matrix whose diagonal elements are the eigenvalues, λ_i, of T and Z is an orthogonal matrix whose columns are the eigenvectors, z_i, of T. Thus

$$Tz_i = \lambda_i z_i, \quad i = 1, 2, \ldots, n.$$

The function may also be used to compute all the eigenvalues and eigenvectors of a complex full, or banded, Hermitian matrix A which has been reduced to real tridiagonal form T as

$$A = QTQ^{\mathrm{H}},$$

where Q is unitary. The spectral factorization of A is then given by

$$A = (QZ)\Lambda(QZ)^{\mathrm{H}}.$$

In this case Q must be formed explicitly and passed to nag_lapack_zstedc (f08jv) in the array **z**, and the function called with **compz** = 'V'. Functions which may be called to form T and Q are

full matrix	f08fs and f08ft
full matrix, packed storage	f08gs and f08gt
band matrix	f08hs, with **vect** = 'V'

When only eigenvalues are required then this function calls nag_lapack_dsterf (f08jf) to compute the eigenvalues of the tridiagonal matrix T, but when eigenvectors of T are also required and the matrix is not too small, then a divide and conquer method is used, which can be much faster than nag_lapack_zsteqr (f08js), although more storage is required.

4 References

Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J J, Du Croz J J, Greenbaum A, Hammarling S, McKenney A and Sorensen D (1999) *LAPACK Users' Guide* (3rd Edition) SIAM, Philadelphia http://www.netlib.org/lapack/lug

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

5 Parameters

5.1 Compulsory Input Parameters

1: **compz** – CHARACTER(1)

Indicates whether the eigenvectors are to be computed.

compz = 'N'
Only the eigenvalues are computed (and the array **z** is not referenced).

compz = 'V'
The eigenvalues and eigenvectors of A are computed (and the array **z** must contain the matrix Q on entry).

compz = 'I'
The eigenvalues and eigenvectors of T are computed (and the array **z** is initialized by the function).

Constraint: **compz** = 'N', 'V' or 'I'.

2: **d**(:) – REAL (KIND=nag_wp) array

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

The diagonal elements of the tridiagonal matrix.

3: **e**(:) – REAL (KIND=nag_wp) array

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

The subdiagonal elements of the tridiagonal matrix.

4: **z**(ldz, :) – COMPLEX (KIND=nag_wp) array

The first dimension, ldz, of the array **z** must satisfy

if **compz** = 'V' or 'I', $ldz \geq \max(1, \mathbf{n})$;
otherwise $ldz \geq 1$.

The second dimension of the array **z** must be at least $\max(1, \mathbf{n})$ if **compz** = 'V' or 'I', and at least 1 otherwise.

If **compz** = 'V', **z** must contain the unitary matrix Q used in the reduction to tridiagonal form.

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the dimension of the array **d**.

n, the order of the symmetric tridiagonal matrix T.

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

5.3 Output Parameters

1: **d**(:) – REAL (KIND=nag_wp) array

The dimension of the array **d** will be $\max(1, \mathbf{n})$

If **info** = 0, the eigenvalues in ascending order.

2: **e**(:) – REAL (KIND=nag_wp) array

The dimension of the array **e** will be $\max(1, \mathbf{n} - 1)$

3: **z**(ldz, :) – COMPLEX (KIND=nag_wp) array

The first dimension, ldz, of the array **z** will be

if **compz** = 'V' or 'I', $ldz = \max(1, \mathbf{n})$;
otherwise $ldz = 1$.

The second dimension of the array **z** will be $\max(1, \mathbf{n})$ if **compz** = 'V' or 'I' and 1 otherwise.

If **compz** = 'V', **z** contains the orthonormal eigenvectors of the original Hermitian matrix A, and if **compz** = 'I', **z** contains the orthonormal eigenvectors of the symmetric tridiagonal matrix T.

If **compz** = 'N', **z** is not referenced.

4: **info** – INTEGER

info = 0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

info = $-i$

If **info** = $-i$, parameter i had an illegal value on entry. The parameters are numbered as follows:

1: **compz**, 2: **n**, 3: **d**, 4: **e**, 5: **z**, 6: **ldz**, 7: **work**, 8: **lwork**, 9: **rwork**, 10: **lrwork**, 11: **iwork**, 12: **liwork**, 13: **info**.

It is possible that **info** refers to a parameter that is omitted from the MATLAB interface. This usually indicates that an error in one of the other input parameters has caused an incorrect value to be inferred.

info > 0

The algorithm failed to compute an eigenvalue while working on the submatrix lying in rows and columns **info**$/(\mathbf{n} + 1)$ through **info** mod $(\mathbf{n} + 1)$.

7 Accuracy

The computed eigenvalues and eigenvectors are exact for a nearby matrix $(T + E)$, where

$$\|E\|_2 = O(\epsilon)\|T\|_2,$$

and ϵ is the ***machine precision***.

If λ_i is an exact eigenvalue and $\tilde{\lambda}_i$ is the corresponding computed value, then

$$\left|\tilde{\lambda}_i - \lambda_i\right| \leq c(n)\epsilon\|T\|_2,$$

where $c(n)$ is a modestly increasing function of n.

If z_i is the corresponding exact eigenvector, and $\tilde{z}_i$ is the corresponding computed eigenvector, then the angle $\theta(\tilde{z}_i, z_i)$ between them is bounded as follows:

$$\theta(\tilde{z}_i, z_i) \leq \frac{c(n)\epsilon\|T\|_2}{\min\limits_{i \neq j}\left|\lambda_i - \lambda_j\right|}.$$

Thus the accuracy of a computed eigenvector depends on the gap between its eigenvalue and all the other eigenvalues.

See Section 4.7 of Anderson *et al.* (1999) for further details. See also nag_lapack_ddisna (f08fl).

8 Further Comments

If only eigenvalues are required, the total number of floating-point operations is approximately proportional to n^2. When eigenvectors are required the number of operations is bounded above by approximately the same number of operations as nag_lapack_zsteqr (f08js), but for large matrices nag_lapack_zstedc (f08jv) is usually much faster.

The real analogue of this function is nag_lapack_dstedc (f08jh).

9 Example

This example finds all the eigenvalues and eigenvectors of the Hermitian band matrix

$$A = \begin{pmatrix} -3.13 & 1.94 - 2.10i & -3.40 + 0.25i & 0 \\ 1.94 + 2.10i & -1.91 & -0.82 - 0.89i & -0.67 + 0.34i \\ -3.40 - 0.25i & -0.82 + 0.89i & -2.87 & -2.10 - 0.16i \\ 0 & -0.67 - 0.34i & -2.10 + 0.16i & 0.50 \end{pmatrix}.$$

A is first reduced to tridiagonal form by a call to nag_lapack_zhbtrd (f08hs).

9.1 Program Text

```
function f08jv_example

fprintf('f08jv example results\n\n');

% A is Hermitian banded matrix
n = 4;
k = nag_int(2);
a = [ -3.13 + 0.00i,  1.94 - 2.10i, -3.40 + 0.25i,  0    + 0i;
       1.94 + 2.10i, -1.91 + 0.00i, -0.82 - 0.89i, -0.67 + 0.34i;
      -3.40 - 0.25i, -0.82 + 0.89i, -2.87 + 0.00i, -2.10 - 0.16i;
       0    + 0i,    -0.67 - 0.34i, -2.10 + 0.16i,  0.50 + 0i    ];

% Convert to general banded format ...
kl = nag_int(k);
[~, abn, ifail] = f01zd( ...
                         'P', k, k, a, complex(zeros(k+k+1,n)));
% ... and chop to give 'Upper' Hermitian banded format
ab = abn(1:k+1,1:n);

% Reduce A to tridiagonal form A = ZTZ^H
compz = 'V';
uplo  = 'Upper';
[~, Td, Te, ZTZ, info] = f08hs( ...
                                compz, uplo, k, ab, complex(zeros(n, n)));

% Eigenvalues and vectors of A from tridiagonal form (ZTZ, Td, Te).
[w, ~, z, info] = f08jv( ...
                         compz, Td, Te, ZTZ);

% Normalize: largest elements are real
for i = 1:n
  [~,k] = max(abs(real(z(:,i)))+abs(imag(z(:,i))));
  z(:,i) = z(:,i)*conj(z(k,i))/abs(z(k,i));
end

disp(' Eigenvalues of A:');
disp(w');
disp(' Corresponding eigenvectors:');
disp(z);
```

9.2 Program Results

```
     f08jv example results

 Eigenvalues of A:
   -7.0042   -4.0038    0.5968    3.0012

 Corresponding eigenvectors:
   0.7293 + 0.0000i  -0.2128 + 0.1511i  -0.3354 - 0.1604i   0.4732 + 0.1947i
  -0.1654 - 0.2046i   0.7316 + 0.0000i  -0.2804 - 0.3413i   0.0891 + 0.4387i
   0.6081 + 0.0301i   0.3910 - 0.3843i  -0.0144 + 0.1532i  -0.5172 - 0.1938i
   0.1653 - 0.0303i   0.2775 - 0.1378i   0.8019 + 0.0000i   0.4824 + 0.0000i
```