NAG Toolbox

nag_lapack_zheevr (f08fr)

1 Purpose

nag_lapack_zheevr (f08fr) computes selected eigenvalues and, optionally, eigenvectors of a complex n by n Hermitian matrix A. Eigenvalues and eigenvectors can be selected by specifying either a range of values or a range of indices for the desired eigenvalues.

2 Syntax

```
[a, m, w, z, isuppz, info] = nag_lapack_zheevr(jobz, range, uplo, a, vl, vu, il, iu, abstol, 'n', n)
[a, m, w, z, isuppz, info] = f08fr(jobz, range, uplo, a, vl, vu, il, iu, abstol, 'n', n)
```

3 Description

The Hermitian matrix is first reduced to a real tridiagonal matrix T, using unitary similarity transformations. Then whenever possible, nag_lapack_zheevr (f08fr) computes the eigenspectrum using Relatively Robust Representations. nag_lapack_zheevr (f08fr) computes eigenvalues by the dqds algorithm, while orthogonal eigenvectors are computed from various 'good' LDL^{T} representations (also known as Relatively Robust Representations). Gram–Schmidt orthogonalization is avoided as far as possible. More specifically, the various steps of the algorithm are as follows. For the ith unreduced block of T:

(a) compute $T - \sigma_i I = L_i D_i L_i^{\mathrm{T}}$, such that $L_i D_i L_i^{\mathrm{T}}$ is a relatively robust representation,

(b) compute the eigenvalues, λ_j, of $L_i D_i L_i^{\mathrm{T}}$ to high relative accuracy by the dqds algorithm,

(c) if there is a cluster of close eigenvalues, 'choose' σ_i close to the cluster, and go to (a),

(d) given the approximate eigenvalue λ_j of $L_i D_i L_i^{\mathrm{T}}$, compute the corresponding eigenvector by forming a rank-revealing twisted factorization.

The desired accuracy of the output can be specified by the argument **abstol**. For more details, see Dhillon (1997) and Parlett and Dhillon (2000).

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

Barlow J and Demmel J W (1990) Computing accurate eigensystems of scaled diagonally dominant matrices *SIAM J. Numer. Anal.* **27** 762–791

Demmel J W and Kahan W (1990) Accurate singular values of bidiagonal matrices *SIAM J. Sci. Statist. Comput.* **11** 873–912

Dhillon I (1997) A new $O(n^2)$ algorithm for the symmetric tridiagonal eigenvalue/eigenvector problem *Computer Science Division Technical Report No. UCB//CSD-97-971* UC Berkeley

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

Parlett B N and Dhillon I S (2000) Relatively robust representations of symmetric tridiagonals *Linear Algebra Appl.* **309** 121–151

5 Parameters

5.1 Compulsory Input Parameters

1: **jobz** – CHARACTER(1)

Indicates whether eigenvectors are computed.

jobz = 'N'
Only eigenvalues are computed.

jobz = 'V'
Eigenvalues and eigenvectors are computed.

Constraint: **jobz** = 'N' or 'V'.

2: **range** – CHARACTER(1)

If **range** = 'A', all eigenvalues will be found.

If **range** = 'V', all eigenvalues in the half-open interval $(\mathbf{vl}, \mathbf{vu}]$ will be found.

If **range** = 'I', the **il**th to **iu**th eigenvalues will be found.

For **range** = 'V' or 'I' and $\mathbf{iu} - \mathbf{il} < \mathbf{n} - 1$, nag_lapack_dstebz (f08jj) and nag_lapack_zstein (f08jx) are called.

Constraint: **range** = 'A', 'V' or 'I'.

3: **uplo** – CHARACTER(1)

If **uplo** = 'U', the upper triangular part of A is stored.

If **uplo** = 'L', the lower triangular part of A is stored.

Constraint: **uplo** = 'U' or 'L'.

4: **a**$(lda, :)$ – COMPLEX (KIND=nag_wp) array

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

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

The n by n Hermitian matrix A.

If **uplo** = 'U', the upper triangular part of a must be stored and the elements of the array below the diagonal are not referenced.

If **uplo** = 'L', the lower triangular part of a must be stored and the elements of the array above the diagonal are not referenced.

5: **vl** – REAL (KIND=nag_wp)
6: **vu** – REAL (KIND=nag_wp)

If **range** = 'V', the lower and upper bounds of the interval to be searched for eigenvalues.

If **range** = 'A' or 'I', **vl** and **vu** are not referenced.

Constraint: if **range** = 'V', **vl** < **vu**.

7: **il** – INTEGER
8: **iu** – INTEGER

If **range** = 'I', the indices (in ascending order) of the smallest and largest eigenvalues to be returned.

If **range** = 'A' or 'V', **il** and **iu** are not referenced.

Constraints:

if **range** = 'I' and **n** = 0, **il** = 1 and **iu** = 0;
if **range** = 'I' and **n** > 0, 1 ≤ **il** ≤ **iu** ≤ **n**.

9: **abstol** – REAL (KIND=nag_wp)

The absolute error tolerance for the eigenvalues. An approximate eigenvalue is accepted as converged when it is determined to lie in an interval $[a, b]$ of width less than or equal to

$$\textbf{abstol} + \epsilon \max(|a|, |b|),$$

where ϵ is the ***machine precision***. If **abstol** is less than or equal to zero, then $\epsilon \|T\|_1$ will be used in its place, where T is the real tridiagonal matrix obtained by reducing A to tridiagonal form. See Demmel and Kahan (1990).

If high relative accuracy is important, set **abstol** to x02am( ), although doing so does not currently guarantee that eigenvalues are computed to high relative accuracy. See Barlow and Demmel (1990) for a discussion of which matrices can define their eigenvalues to high relative accuracy.

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the first dimension of the array **a** and the second dimension of the array **a**. (An error is raised if these dimensions are not equal.)

n, the order of the matrix A.

Constraint: **n** ≥ 0.

5.3 Output Parameters

1: **a**$(lda, :)$ – COMPLEX (KIND=nag_wp) array

The first dimension of the array **a** will be $\max(1, \textbf{n})$.

The second dimension of the array **a** will be $\max(1, \textbf{n})$.

The lower triangle (if **uplo** = 'L') or the upper triangle (if **uplo** = 'U') of **a**, including the diagonal, is overwritten.

2: **m** – INTEGER

The total number of eigenvalues found. 0 ≤ **m** ≤ **n**.

If **range** = 'A', **m** = **n**.

If **range** = 'I', **m** = **iu** − **il** + 1.

3: **w**$(:)$ – REAL (KIND=nag_wp) array

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

The first **m** elements contain the selected eigenvalues in ascending order.

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

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

if **jobz** = 'V', $ldz = \max(1, \textbf{n})$;
otherwise $ldz = 1$.

The second dimension of the array **z** will be $\max(1, \textbf{m})$ if **jobz** = 'V' and 1 otherwise.

If **jobz** = 'V', the first **m** columns of Z contain the orthonormal eigenvectors of the matrix A corresponding to the selected eigenvalues, with the ith column of Z holding the eigenvector associated with $\mathbf{w}(i)$.

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

5: **isuppz**(:) – INTEGER array

The dimension of the array **isuppz** will be $\max(1, 2 \times \mathbf{m})$

The support of the eigenvectors in **z**, i.e., the indices indicating the nonzero elements in **z**. The ith eigenvector is nonzero only in elements $\mathbf{isuppz}(2 \times i - 1)$ through $\mathbf{isuppz}(2 \times i)$. Implemented only for **range** = 'A' or 'I' and $\mathbf{iu} - \mathbf{il} = \mathbf{n} - 1$.

6: **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: **jobz**, 2: **range**, 3: **uplo**, 4: **n**, 5: **a**, 6: **lda**, 7: **vl**, 8: **vu**, 9: **il**, 10: **iu**, 11: **abstol**, 12: **m**, 13: **w**, 14: **z**, 15: **ldz**, 16: **isuppz**, 17: **work**, 18: **lwork**, 19: **rwork**, 20: **lrwork**, 21: **iwork**, 22: **liwork**, 23: **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

nag_lapack_zheevr (f08fr) failed to converge.

7 Accuracy

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

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

and ϵ is the ***machine precision***. See Section 4.7 of Anderson *et al.* (1999) for further details.

8 Further Comments

The total number of floating-point operations is proportional to n^3.

The real analogue of this function is nag_lapack_dsyevr (f08fd).

9 Example

This example finds the eigenvalues with indices in the range $[2, 3]$, and the corresponding eigenvectors, of the Hermitian matrix

$$A = \begin{pmatrix} 1 & 2-i & 3-i & 4-i \\ 2+i & 2 & 3-2i & 4-2i \\ 3+i & 3+2i & 3 & 4-3i \\ 4+i & 4+2i & 4+3i & 4 \end{pmatrix}.$$

Information on required and provided workspace is also output.

9.1 Program Text

```
     function f08fr_example

fprintf('f08fr example results\n\n');

% Upper triangular part of Hermitian matrix A
uplo  = 'Upper';
n = 4;
a = [ 1 + 0i,  2 - 1i,  3 - 1i,  4 - 1i;
      0 + 0i,  2 + 0i,  3 - 2i,  4 - 2i;
      0 + 0i,  0 + 0i,  3 + 0i,  4 - 3i;
      0 + 0i,  0 + 0i,  0 + 0i,  4 + 0i];

% Compute 2nd and 3rd eigenvalues and coresponding vectors
jobz  = 'Vectors';
range = 'I';
vl = 0;
vu = 0;
il = nag_int(2);
iu = nag_int(3);
abstol = 0;
[~, m, w, z, ~, info] = ...
  f08fr( ...
         jobz, range, uplo, a, vl, vu, il, iu, abstol);

% Normalize vectors, largest element is real and positive.
for i = 1:m
  [~,k] = max(abs(real(z(:,i)))+abs(imag(z(:,i))));
  z(:,i) = z(:,i)*conj(z(k,i))/abs(z(k,i));
end

% Display
fprintf(' Eigenvalues numbered 2 to 3 are:\n');
disp(w(1:m));

ncols  = nag_int(80);
indent = nag_int(0);
[ifail] = x04db( ...
                 'General', ' ', z, 'Bracketed', 'F7.4', ...
                 'Corresponding eigenvectors', 'Integer', 'Integer', ...
                 ncols, indent);
```

9.2 Program Results

```
     f08fr example results

 Eigenvalues numbered 2 to 3 are:
   -0.6886
    1.1412

 Corresponding eigenvectors
                     1                  2
 1  ( 0.6470, 0.0000) ( 0.0179,-0.4453)
 2  (-0.4984,-0.1130) ( 0.5706, 0.0000)
 3  ( 0.2949, 0.3165) (-0.1530, 0.5273)
 4  (-0.2241,-0.2878) (-0.2118,-0.3598)
```