

NAG Toolbox for MATLAB

f08fp

1 Purpose

f08fp 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, jfail, info] = f08fp(jobz, range, uplo, a, vl, vu, il, iu, abstol, 'n', n)
```

3 Description

The Hermitian matrix A is first reduced to real tridiagonal form, using unitary similarity transformations. The required eigenvalues and eigenvectors are then computed from the tridiagonal matrix; the method used depends upon whether all, or selected, eigenvalues and eigenvectors are 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 URL: <http://www.netlib.org/lapack/lug>

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

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: **jobz** – string

If **jobz** = 'N', compute eigenvalues only.

If **jobz** = 'V', compute eigenvalues and eigenvectors.

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

2: **range** – string

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

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

If **range** = 'I', the **ilth** to **iuth** eigenvalues will be found.

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

3: **uplo** – string

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 array**

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

The second dimension of the array 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 – double scalar**6: **vu – double scalar**

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 – int32 scalar**8: **iu – int32 scalar**

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 $\mathbf{n} = 0$, **il** = 1 and **iu** = 0;

if $\mathbf{n} > 0$, $1 \leq \mathbf{il} \leq \mathbf{iu} \leq \mathbf{n}$.

9: **abstol – double scalar**

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

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

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 tridiagonal matrix obtained by reducing A to tridiagonal form. Eigenvalues will be computed most accurately when **abstol** is set to twice the underflow threshold $2 \times \text{x02am}()$, not zero. If this function returns with **info** > 0, indicating that some eigenvectors did not converge, try setting **abstol** to $2 \times \text{x02am}()$. See Demmel and Kahan 1990.

5.2 Optional Input Parameters

1: **n – int32 scalar**

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

5.3 Input Parameters Omitted from the MATLAB Interface

lda, ldz, work, lwork, rwork, iwork

5.4 Output Parameters

1: **a(lda,*) – complex array**

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

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

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

2: **m** – **int32 scalar**

The total number of eigenvalues found.

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

If **range** = 'V', the exact value of **m** is not known in advance, but will satisfy $0 \leq \mathbf{m} \leq \mathbf{n}$.

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

3: **w(*)** – **double array**

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

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

4: **z(ldz,*)** – **complex array**

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

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

The second dimension of the array must be at least $\max(1, \mathbf{m})$ if **jobz** = 'V', and at least 1 otherwise

If **jobz** = 'V', then if **info** = 0, the first **m** columns of **Z** contain the orthonormal eigenvectors of the matrix **A** corresponding to the selected eigenvalues, with the *i*th column of **Z** holding the eigenvector associated with **w**(*i*).

If an eigenvector fails to converge, then that column of **Z** contains the latest approximation to the eigenvector, and the index of the eigenvector is returned in **jfail**.

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

Note: you must ensure that at least $\max(1, \mathbf{m})$ columns are supplied in the array **z**; if **range** = 'V', the exact value of **m** is not known in advance and an upper bound must be used.

5: **jfail(*)** – **int32 array**

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

If **jobz** = 'V', then if **info** = 0, the first **m** elements of **jfail** are zero.

If **info** > 0, **jfail** contains the indices of the eigenvectors that failed to converge.

If **jobz** = 'E', **jfail** is not referenced.

6: **info** – **int32 scalar**

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

6 Error Indicators and Warnings

Errors or warnings detected by the function:

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: **work**, 17: **lwork**, 18: **rwork**, 19: **iwork**, 20: **jfail**, 21: **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

If **info** = i , then i eigenvectors failed to converge. Their indices are stored in array **jfail**. Please see **abstol**.

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 f08fb.

9 Example

```

jobz = 'Vectors';
range = 'Values in range';
uplo = 'Upper';
a = [complex(1, +0), complex(2, -1), complex(3, -1), complex(4, -1);
      complex(0, 0), complex(2, +0), complex(3, -2), complex(4, -2);
      complex(0, 0), complex(0, 0), complex(3, +0), complex(4, -3);
      complex(0, 0), complex(0, 0), complex(0, 0), complex(4, +0)];
vl = -2;
vu = 2;
il = int32(134537040);
iu = int32(0);
abstol = 0;
[aOut, m, w, z, jfail, info] = f08fp(jobz, range, uplo, a, vl, vu, il,
iu, abstol)

```

```

aOut =
    -0.2187          1.0422          0.4448 + 0.4277i    0.3367 +
    0.0008i          0      -0.3942          -3.4564          0.3567 -
    0.0783i          0          0          6.6129          -7.8740
          0          0          0          4.0000
m =
          2
w =
    -0.6886
     1.1412
         0
         0
z =
    -0.3975 + 0.5105i    -0.3746 - 0.2414i
     0.3953 - 0.3238i     0.2895 - 0.4917i
    -0.4309 + 0.0383i     0.3768 + 0.3994i
     0.3648          -0.4175
jfail =
         0
         0
         0
         0

```

info = 0
