NAG Toolbox for MATLAB

f08uq

1 Purpose

f08uq computes all the eigenvalues and, optionally, the eigenvectors of a complex generalized Hermitian-definite banded eigenproblem, of the form

$$Az = \lambda Bz,$$

where A and B are Hermitian and banded, and B is also positive-definite. If eigenvectors are desired, it uses a divide-and-conquer algorithm.

2 Syntax

```
[ab, bb, w, z, info] = f08uq(jobz, uplo, ka, kb, ab, bb, 'n', n)
```

3 Description

The generalized Hermitian-definite band problem

$$Az = \lambda Bz$$

is first reduced to a standard band Hermitian problem

$$Cx = \lambda x,$$

where C is a Hermitian band matrix, using Wilkinson's modification to Crawford's algorithm (see Crawford 1973 and Wilkinson 1977). The Hermitian eigenvalue problem is then solved for the eigenvalues and the eigenvectors, if required, which are then backtransformed to the eigenvectors of the original problem.

The eigenvectors are normalized so that the matrix of eigenvectors, Z, satisfies

$$Z^{\mathrm{H}}AZ = \Lambda \quad \text{and} \quad Z^{\mathrm{H}}BZ = I,$$

where Λ is the diagonal matrix whose diagonal elements are the eigenvalues.

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

Crawford C R 1973 Reduction of a band-symmetric generalized eigenvalue problem *Comm. ACM* **16** 41–44

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

Wilkinson J H 1977 Some recent advances in numerical linear algebra *The State of the Art in Numerical Analysis* (ed D A H Jacobs) Academic Press

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: **uplo – string**

If **uplo** = 'U', the upper triangles of A and B are stored.

If **uplo** = 'L', the lower triangles of A and B are stored.

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

3: **ka – int32 scalar**

If **uplo** = 'U', the number of superdiagonals, k_a, of the matrix A.

If **uplo** = 'L', the number of subdiagonals, k_a, of the matrix A.

Constraint: **ka** ≥ 0.

4: **kb – int32 scalar**

If **uplo** = 'U', the number of superdiagonals, k_b, of the matrix B.

If **uplo** = 'L', the number of subdiagonals, k_b, of the matrix B.

Constraint: **kb** ≥ 0.

5: **ab(ldab,∗) – complex array**

The first dimension of the array **ab** must be at least **ka** $+ 1$

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

The upper or lower triangle of the n by n Hermitian band matrix A.

The matrix is stored in rows 1 to $k_a + 1$, more precisely,

> if **uplo** = 'U', the elements of the upper triangle of A within the band must be stored with element A_{ij} in $\mathbf{ab}(k_a + 1 + i - j, j)$ for $\max(1 j - k_a) \leq i \leq j$;
> if **uplo** = 'L', the elements of the lower triangle of A within the band must be stored with element A_{ij} in $\mathbf{ab}(1 + i - j, j)$ for $j \leq i \leq \min(n j + k_a)$.

6: **bb(ldbb,∗) – complex array**

The first dimension of the array **bb** must be at least **kb** $+ 1$

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

The upper or lower triangle of the n by n Hermitian band matrix B.

The matrix is stored in rows 1 to $k_b + 1$, more precisely,

> if **uplo** = 'U', the elements of the upper triangle of B within the band must be stored with element B_{ij} in $\mathbf{bb}(k_b + 1 + i - j, j)$ for $\max(1 j - k_b) \leq i \leq j$;
> if **uplo** = 'L', the elements of the lower triangle of B within the band must be stored with element B_{ij} in $\mathbf{bb}(1 + i - j, j)$ for $j \leq i \leq \min(n j + k_b)$.

5.2 Optional Input Parameters

1: **n – int32 scalar**

Default: The second dimension of the array **ab** The second dimension of the array **bb**.

n, the order of the matrices A and B.

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

5.3 Input Parameters Omitted from the MATLAB Interface

ldab, ldbb, ldz, work, lwork, rwork, lrwork, iwork, liwork

5.4 Output Parameters

1: **ab(ldab,*) – complex array**

The first dimension of the array **ab** must be at least $\mathbf{ka} + 1$

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

the contents of **ab** are destroyed.

2: **bb(ldbb,*) – complex array**

The first dimension of the array **bb** must be at least $\mathbf{kb} + 1$

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

The factor S from the split Cholesky factorization $B = S^{\mathrm{H}}S$, as returned by f08ut.

3: **w(*) – double array**

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

If $\mathbf{info} = 0$, the eigenvalues in ascending order.

4: **z(ldz,*) – complex array**

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

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

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

If **jobz** = 'V', then if $\mathbf{info} = 0$, **z** contains the matrix Z of eigenvectors, with the ith column of Z holding the eigenvector associated with $\mathbf{w}(i)$. The eigenvectors are normalized so that $Z^{\mathrm{H}}BZ = I$.

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

5: **info – int32 scalar**

$\mathbf{info} = 0$ unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

$\mathbf{info} = -i$

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

1: **jobz**, 2: **uplo**, 3: **n**, 4: **ka**, 5: **kb**, 6: **ab**, 7: **ldab**, 8: **bb**, 9: **ldbb**, 10: **w**, 11: **z**, 12: **ldz**, 13: **work**, 14: **lwork**, 15: **rwork**, 16: **lrwork**, 17: **iwork**, 18: **liwork**, 19: **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** $= i$ and $i \leq$ **n**, the algorithm failed to converge: i off-diagonal elements of an intermediate tridiagonal form did not converge to zero.

If **info** $= i$ and $i >$ **n**, if **info** $=$ **n** $+ i$, for $1 \leq i \leq$ **n**, then f08ut returned **info** $= i$: B is not positive-definite. The factorization of B could not be completed and no eigenvalues or eigenvectors were computed.

7 Accuracy

If B is ill-conditioned with respect to inversion, then the error bounds for the computed eigenvalues and vectors may be large, although when the diagonal elements of B differ widely in magnitude the eigenvalues and eigenvectors may be less sensitive than the condition of B would suggest. See Section 4.10 of Anderson *et al.* 1999 for details of the error bounds.

8 Further Comments

The total number of floating-point operations is proportional to n^3 if **jobz** $=$ 'V' and, assuming that $n \gg k_a$, is approximately proportional to $n^2 k_a$ otherwise.

The real analogue of this function is f08uc.

9 Example

```
jobz = 'No vectors';
uplo = 'U';
ka = int32(2);
kb = int32(1);
ab = [complex(0, 0), complex(0, 0), complex(-1.4, +0.25), complex(-0.67,
+0.34);
     complex(0, 0), complex(1.94, -2.1), complex(-0.82, -0.89), complex(-
1.1, -0.16);
          complex(-1.13,  +0),  complex(-1.91,  +0),  complex(-1.87,  +0),
complex(0.5, +0)];
bb   =  [complex(0,  +0),  complex(1.08,  -1.73),  complex(-0.04,  +0.29),
complex(-0.33, +2.24);
             complex(9.89,  +0),  complex(1.69,  +0),  complex(2.65,  +0),
complex(2.17, +0)];
[abOut, bbOut, w, z, info] = f08uq(jobz, uplo, ka, kb, ab, bb)
```

```
abOut =
         0                      0              -0.8302 + 0.1482i   -0.8561 +
1.5855i
        0             1.5456             2.7633             2.4855
  -0.1143            -1.7545            -3.7946            -1.0557
bbOut =
         0                 0.3434 - 0.5501i   -0.0746 + 0.5408i   -0.2240 +
1.5206i
   3.1448             0.9856             0.5362             1.4731
w =
   -6.6089
   -2.0416
    0.1603
    1.7712
z =
   1.0e-53 *
   0.0000 + 0.0000i   0.0000 + 0.0000i  -0.1687                 0
info =
           0
```