NAG Toolbox for MATLAB

f08up

1 Purpose

f08up 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. Eigenvalues and eigenvectors can be selected by specifying either all eigenvalues, a range of values or a range of indices for the desired eigenvalues.

2 Syntax

```
[ab, bb, q, m, w, z, jfail, info] = f08up(jobz, range, uplo, ka, kb, ab,
bb, vl, vu, il, iu, abstol, '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 required eigenvalues and eigenvectors, and the eigenvectors are then backtransformed to the eigenvectors of the original problem.

The eigenvectors are normalized so that

$$z^{\mathrm{H}}Az = \lambda \quad \text{and} \quad z^{\mathrm{H}}Bz = 1.$$

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

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

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: **range – string**

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.

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

3: **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'.

4: **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: $\mathbf{ka} \geq 0$.

5: **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: $\mathbf{kb} \geq 0$.

6: **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 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 $\max(1j - k_a) \leq i \leq j$;
> 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 $j \leq i \leq \min(nj + k_a)$.

7: **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 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 $\max(1j - k_b) \leq i \leq j$;
> 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 $j \leq i \leq \min(nj + k_b)$.

8: **vl – double scalar**

9: **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**.

10: **il – int32 scalar**

11: **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 **n** = 0, **il** = 1 and **iu** = 0;
> if **n** > 0, $1 \leq$ **il** $\leq$ **iu** $\leq$ **n**.

12: **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 C to tridiagonal form. Eigenvalues will be computed most accurately when **abstol** is set to twice the underflow threshold 2 × x02am( ), not zero. If this function returns with **info** > 0, indicating that some eigenvectors did not converge, try setting **abstol** to 2 × x02am( ). See Demmel and Kahan 1990.

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: **n** ≥ 0.

5.3 Input Parameters Omitted from the MATLAB Interface

ldab, ldbb, ldq, ldz, work, rwork, iwork

5.4 Output Parameters

1: **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, **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: **q(ldq,$*$) – complex array**

The first dimension, **ldq**, of the array **q** must satisfy

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

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

If **jobz** = 'V', the n by n matrix used in the reduction of $Az = \lambda Bz$ to standard form, i.e., $Cx = \lambda x$, and subsequently C to tridiagonal form.

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

4: **m – int32 scalar**

The total number of eigenvalues found.

If **range** = 'A', $\mathbf{m} = \mathbf{n}$.

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', $\mathbf{m} = \mathbf{iu} - \mathbf{il} + 1$.

5: **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.

6: **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.

7: **jfail($*$) – int32 array**

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

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

If $\mathbf{info} > 0$, **jfail** contains the indices of the eigenvectors that failed to converge.

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

8: **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:

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: **ka**, 6: **kb**, 7: **ab**, 8: **ldab**, 9: **bb**, 10: **ldbb**, 11: **q**, 12: **ldq**, 13: **vl**, 14: **vu**, 15: **il**, 16: **iu**, 17: **abstol**, 18: **m**, 19: **w**, 20: **z**, 21: **ldz**, 22: **work**, 23: **rwork**, 24: **iwork**, 25: **jfail**, 26: **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 $= 1$ to N

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

info $>$ N

f08uf returned an error code; i.e., if **info** $=$ **n** $+ i$, for $1 \leq i \leq$ **n**, then the leading minor of order i of 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 **range** = 'A', and assuming that $n \gg k_a$, is approximately proportional to $n^2 k_a$ if **jobz** = 'N'. Otherwise the number of floating-point operations depends upon the number of eigenvectors computed.

The real analogue of this function is f08ub.

9 Example

```
jobz = 'Vectors';
range = 'Values in range';
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)];
vl = 0;
vu = 2;
il = int32(0);
```

```
iu = int32(8185080);
abstol = 0;
[abOut, bbOut, q, m, w, z, jfail, info] = ...
    f08up(jobz, range, uplo, ka, kb, ab, bb, vl, vu, il, iu, abstol)
```

```
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
q =
    0.3180               -0.1735 - 0.0251i  -0.0358 + 0.0788i   0.0357 -
0.0653i
         0                0.3424 + 0.7783i   0.4161 - 0.0551i  -0.3604 +
0.0208i
         0               -1.7389 + 0.2747i   0.6388 - 0.6501i  -0.5894 +
0.5169i
         0               -0.5480 - 1.7533i   1.0400 + 0.2112i  -0.2779 -
0.9114i
m =
           2
w =
    0.1603
    1.7712
         0
         0
z =
   0.1908 + 0.0137i   0.0494 - 0.0045i        0                  0
   0.1413 + 0.1012i   0.2505 + 0.4427i        0                  0
  -0.0437 - 0.0905i  -0.9705 + 0.0679i        0                  0
  -0.2135 + 0.2880i   0.0606 - 1.3227i        0                  0
jfail =
           0
           0
           0
           0
info =
           0
```