NAG Toolbox for MATLAB

f08ss

1 Purpose

f08ss reduces a complex Hermitian-definite generalized eigenproblem $Az = \lambda Bz$, $ABz = \lambda z$ or $BAz = \lambda z$ to the standard form $Cy = \lambda y$, where A is a complex Hermitian matrix and B has been factorized by f07fr.

2 Syntax

```
[a, info] = f08ss(itype, uplo, a, b, 'n', n)
```

3 Description

To reduce the complex Hermitian-definite generalized eigenproblem $Az = \lambda Bz$, $ABz = \lambda z$ or $BAz = \lambda z$ to the standard form $Cy = \lambda y$, f08ss must be preceded by a call to f07fr which computes the Cholesky factorization of B; B must be positive-definite.

The different problem types are specified by the parameter **itype**, as indicated in the table below. The table shows how C is computed by the function, and also how the eigenvectors z of the original problem can be recovered from the eigenvectors of the standard form.

itype	Problem	**uplo**	B	C	z
1	$Az = \lambda Bz$	'U' 'L'	$U^H U$ LL^H	$U^{-H} A U^{-1}$ $L^{-1} A L^{-H}$	$U^{-1} y$ $L^{-H} y$
2	$ABz = \lambda z$	'U' 'L'	$U^H U$ LL^H	UAU^H $L^H AL$	$U^{-1} y$ $L^{-H} y$
3	$BAz = \lambda z$	'U' 'L'	$U^H U$ LL^H	UAU^H $L^H AL$	$U^H y$ Ly

4 References

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: **itype – int32 scalar**

Indicates how the standard form is computed.

itype $= 1$

if **uplo** = 'U', $C = U^{-H} A U^{-1}$;
if **uplo** = 'L', $C = L^{-1} A L^{-H}$.

itype $= 2$ or 3

if **uplo** = 'U', $C = UAU^H$;
if **uplo** = 'L', $C = L^H AL$.

Constraint: **itype** $= 1$, 2 or 3.

2: **uplo – string**

Indicates whether the upper or lower triangular part of A is stored and how B has been factorized.

uplo = 'U'

The upper triangular part of A is stored and $B = U^{\mathrm{H}}U$.

uplo = 'L'

The lower triangular part of A is stored and $B = LL^{\mathrm{H}}$.

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

3: **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.

4: **b(ldb,∗) – complex array**

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

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

The Cholesky factor of B as specified by **uplo** and returned by f07fr.

5.2 Optional Input Parameters

1: **n – int32 scalar**

Default: The second dimension of the array **a** The second dimension of the array **b**.

n, the order of the matrices A and B.

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

5.3 Input Parameters Omitted from the MATLAB Interface

lda, ldb

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 upper or lower triangle of **a** contains the corresponding upper or lower triangle of C as specified by **itype** and **uplo**.

2: **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: **itype**, 2: **uplo**, 3: **n**, 4: **a**, 5: **lda**, 6: **b**, 7: **ldb**, 8: **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.

7 Accuracy

Forming the reduced matrix C is a stable procedure. However it involves implicit multiplication by B^{-1} (if **itype** $= 1$) or B (if **itype** $= 2$ or 3). When f08ss is used as a step in the computation of eigenvalues and eigenvectors of the original problem, there may be a significant loss of accuracy if B is ill-conditioned with respect to inversion. See the document for f08sn for further details.

8 Further Comments

The total number of real floating-point operations is approximately $4n^3$.

The real analogue of this function is f08se.

9 Example

```
uplo = 'L';
itype = int32(1);
uplo = 'L';
a = [complex(-7.36, +0), complex(0, 0), complex(0, 0), complex(0, 0);
      complex(0.77, +0.43), complex(3.49, +0), complex(0, 0), complex(0,
0);
       complex(-0.64, +0.92), complex(2.19, -4.45), complex(0.12, +0),
complex(0, 0);
       complex(3.01, +6.97), complex(1.9, -3.73), complex(2.88, +3.17),
complex(-2.54, +0)];
b = [complex( 3.23, 0.00), complex(0, 0), complex(0, 0), complex(0, 0);
  complex( 1.51, 1.92), complex( 3.58, 0.00), complex(0, 0), complex(0,
0);
   complex(  1.90,-0.84),  complex(-0.23,-1.11),  complex(  4.09,  0.00),
complex(0, 0);
   complex(  0.42,-2.50),  complex(-1.18,-1.37),  complex(  2.33,  0.14),
complex( 4.29, 0.00)];
[b, info] = f07fr(uplo, b);
[aOut, info] = f08ss(itype, uplo, a, b)
```

```
aOut =
  -2.2786                  0                  0                  0
   1.7799 + 2.0310i  -1.1255                  0                  0
   2.2594 - 0.0996i   0.0090 - 0.4261i  -0.3715                  0
  -0.1206 - 2.5286i  -1.0602 - 0.8600i   2.3103 + 0.9198i  -0.7133
info =
           0
```