NAG Toolbox

nag_lapack_dsygst (f08se)

1 Purpose

nag_lapack_dsygst (f08se) reduces a real symmetric-definite generalized eigenproblem $Az = \lambda Bz$, $ABz = \lambda z$ or $BAz = \lambda z$ to the standard form $Cy = \lambda y$, where A is a real symmetric matrix and B has been factorized by nag_lapack_dpotrf (f07fd).

2 Syntax

```
[a, info] = nag_lapack_dsygst(itype, uplo, a, b, 'n', n)
[a, info] = f08se(itype, uplo, a, b, 'n', n)
```

3 Description

To reduce the real symmetric-definite generalized eigenproblem $Az = \lambda Bz$, $ABz = \lambda z$ or $BAz = \lambda z$ to the standard form $Cy = \lambda y$, nag_lapack_dsygst (f08se) must be preceded by a call to nag_lapack_dpotrf (f07fd) which computes the Cholesky factorization of B; B must be positive definite.

The different problem types are specified by the argument **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'	$U^{\mathrm{T}}U$ LL^{T}	$U^{-\mathrm{T}}AU^{-1}$ $L^{-1}AL^{-\mathrm{T}}$	$U^{-1}y$ $L^{-\mathrm{T}}y$
2	$ABz = \lambda z$	\`U' \`L'	$U^{\mathrm{T}}U$ LL^{T}	UAU^{T} $L^{\mathrm{T}}AL$	$U^{-1}y$ $L^{-\mathrm{T}}y$
3	$BAz = \lambda z$	\`U' \`L'	$U^{\mathrm{T}}U$ LL^{T}	UAU^{T} $L^{\mathrm{T}}AL$	$U^{\mathrm{T}}y$ 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** – INTEGER

Indicates how the standard form is computed.

itype $= 1$

if **uplo** $=$ 'U', $C = U^{-\mathrm{T}}AU^{-1}$;

if **uplo** $=$ 'L', $C = L^{-1}AL^{-\mathrm{T}}$.

itype $= 2$ or 3

if **uplo** $=$ 'U', $C = UAU^{\mathrm{T}}$;

if **uplo** $=$ 'L', $C = L^{\mathrm{T}}AL$.

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

2: **uplo** – CHARACTER(1)

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 $B = U^{\mathrm{T}}U$.

uplo $=$ 'L'
The lower triangular part of A is stored and $B = LL^{\mathrm{T}}$.

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

3: **a**$(lda,:)$ – REAL (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 symmetric 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**$(ldb,:)$ – REAL (KIND=nag_wp) array

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

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

The Cholesky factor of B as specified by **uplo** and returned by nag_lapack_dpotrf (f07fd).

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the first dimension of the arrays **a**, **b** and the second dimension of the arrays **a**, **b**.

n, the order of the matrices A and B.

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

5.3 Output Parameters

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

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

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

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

2: **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: **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 nag_lapack_dsygst (f08se) 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 nag_lapack_dsygv (f08sa) for further details.

8 Further Comments

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

The complex analogue of this function is nag_lapack_zhegst (f08ss).

9 Example

This example computes all the eigenvalues of $Az = \lambda Bz$, where

$$A = \begin{pmatrix} 0.24 & 0.39 & 0.42 & -0.16 \\ 0.39 & -0.11 & 0.79 & 0.63 \\ 0.42 & 0.79 & -0.25 & 0.48 \\ -0.16 & 0.63 & 0.48 & -0.03 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 4.16 & -3.12 & 0.56 & -0.10 \\ -3.12 & 5.03 & -0.83 & 1.09 \\ 0.56 & -0.83 & 0.76 & 0.34 \\ -0.10 & 1.09 & 0.34 & 1.18 \end{pmatrix}.$$

Here B is symmetric positive definite and must first be factorized by nag_lapack_dpotrf (f07fd). The program calls nag_lapack_dsygst (f08se) to reduce the problem to the standard form $Cy = \lambda y$; then nag_lapack_dsytrd (f08fe) to reduce C to tridiagonal form, and nag_lapack_dsterf (f08jf) to compute the eigenvalues.

9.1 Program Text

```
     function f08se_example

fprintf('f08se example results\n\n');

% Sove Az = lambda Bz
  % A and B are symmetric, B is positive definite:
a = [ 0.24,  0.39,  0.42, -0.16;
      0.39, -0.11,  0.79,  0.63;
      0.42,  0.79, -0.25,  0.48;
     -0.16,  0.63,  0.48, -0.03];
b = [ 4.16  -3.12   0.56  -0.10;
     -3.12   5.03  -0.83   1.09;
      0.56  -0.83   0.76   0.34;
     -0.10   1.09   0.34   1.18];

% Factorize B
uplo = 'L';
[bfac, info] = f07fd(uplo, b);

% Reduce problem to standard form Cy = lambda*y
itype = nag_int(1);
[c, info] = f08se( ...
```

```
    itype, uplo, a, bfac);

% Find eigenvalues lambda
jobz = 'No Vectors';
[~, w, info] = f08fa( ...
       jobz, uplo, c);

disp('Eigenvalues:');
disp(w);
```

9.2 Program Results

```
     f08se example results

Eigenvalues:
   -2.2254
   -0.4548
    0.1001
    1.1270
```