NAG Toolbox

nag_lapack_dsygvx (f08sb)

1 Purpose

nag_lapack_dsygvx (f08sb) computes selected eigenvalues and, optionally, eigenvectors of a real generalized symmetric-definite eigenproblem, of the form

$$Az = \lambda Bz, \quad ABz = \lambda z \quad \text{or} \quad BAz = \lambda z,$$

where A and B are symmetric and B is also positive definite. 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, b, m, w, z, jfail, info] = nag_lapack_dsygvx(itype, jobz, range, uplo, a, b,
vl, vu, il, iu, abstol, 'n', n)
```

```
[a, b, m, w, z, jfail, info] = f08sb(itype, jobz, range, uplo, a, b, vl, vu, il,
iu, abstol, 'n', n)
```

3 Description

nag_lapack_dsygvx (f08sb) first performs a Cholesky factorization of the matrix B as $B = U^{\mathrm{T}}U$, when **uplo** $=$ 'U' or $B = LL^{\mathrm{T}}$, when **uplo** $=$ 'L'. The generalized problem is then reduced to a standard symmetric eigenvalue problem

$$Cx = \lambda x,$$

which is solved for the desired eigenvalues and eigenvectors; the eigenvectors are then backtransformed to give the eigenvectors of the original problem.

For the problem $Az = \lambda Bz$, the eigenvectors are normalized so that the matrix of eigenvectors, Z, satisfies

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

where Λ is the diagonal matrix whose diagonal elements are the eigenvalues. For the problem $ABz = \lambda z$ we correspondingly have

$$Z^{-1}AZ^{-\mathrm{T}} = \Lambda \quad \text{and} \quad Z^{\mathrm{T}}BZ = I,$$

and for $BAz = \lambda z$ we have

$$Z^{\mathrm{T}}AZ = \Lambda \quad \text{and} \quad Z^{\mathrm{T}}B^{-1}Z = I.$$

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 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: **itype** – INTEGER

Specifies the problem type to be solved.

itype $= 1$
$Az = \lambda Bz$.

itype $= 2$
$ABz = \lambda z$.

itype $= 3$
$BAz = \lambda z$.

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

2: **jobz** – CHARACTER(1)

Indicates whether eigenvectors are computed.

jobz = 'N'
Only eigenvalues are computed.

jobz = 'V'
Eigenvalues and eigenvectors are computed.

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

3: **range** – CHARACTER(1)

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'.

4: **uplo** – CHARACTER(1)

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'.

5: $\mathbf{a}(lda, :)$ – 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.

6: $\mathbf{b}(ldb, :)$ – 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 n by n symmetric matrix B.

If **uplo** $=$ 'U', the upper triangular part of b must be stored and the elements of the array below the diagonal are not referenced.

If **uplo** $=$ 'L', the lower triangular part of b must be stored and the elements of the array above the diagonal are not referenced.

7: **vl** – REAL (KIND=nag_wp)

8: **vu** – REAL (KIND=nag_wp)

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**.

9: **il** – INTEGER

10: **iu** – INTEGER

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 **range** $=$ 'I' and **n** $= 0$, **il** $= 1$ and **iu** $= 0$;
if **range** $=$ 'I' and **n** > 0, $1 \leq$ **il** $\leq$ **iu** $\leq$ **n**.

11: **abstol** – REAL (KIND=nag_wp)

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 $2 \times$ x02am( ), not zero. If this function returns with **info** $= 1$ to **n**, indicating that some eigenvectors did not converge, try setting **abstol** to $2 \times$ x02am( ). See Demmel and Kahan (1990).

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**. (An error is raised if these dimensions are not equal.)

n, the order of the matrices A and B.

Constraint: **n** ≥ 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 lower triangle (if **uplo** $=$ 'L') or the upper triangle (if **uplo** $=$ 'U') of **a**, including the diagonal, is overwritten.

2: **b**$(ldb,:)$ – REAL (KIND=nag_wp) array

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

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

The triangular factor U or L from the Cholesky factorization $B = U^{\mathrm{T}}U$ or $B = LL^{\mathrm{T}}$.

3: **m** – INTEGER

The total number of eigenvalues found. $0 \leq \mathbf{m} \leq \mathbf{n}$.

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

If **range** = 'I', $\mathbf{m} = \mathbf{iu} - \mathbf{il} + 1$.

4: **w**(**n**) – REAL (KIND=nag_wp) array

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

5: **z**(ldz, :) – REAL (KIND=nag_wp) array

The first dimension, ldz, of the array **z** will be

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

The second dimension of the array **z** will be $\max(1, \mathbf{m})$ if **jobz** = 'V' and 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 ith column of Z holding the eigenvector associated with **w**(i). The eigenvectors are normalized as follows:

if **itype** = 1 or 2, $Z^{\mathrm{T}}BZ = I$;

if **itype** = 3, $Z^{\mathrm{T}}B^{-1}Z = I$;

if an eigenvector fails to converge (**info** = 1 to **n**), then that column of Z contains the latest approximation to the eigenvector, and the index of the eigenvector is returned in **jfail**.

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

6: **jfail**(:) – INTEGER array

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

If **jobz** = 'V', then

if **info** = 0, the first **m** elements of **jfail** are zero;

if **info** = 1 to **n**, **jfail** contains the indices of the eigenvectors that failed to converge.

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

7: **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: **jobz**, 3: **range**, 4: **uplo**, 5: **n**, 6: **a**, 7: **lda**, 8: **b**, 9: **ldb**, 10: **vl**, 11: **vu**, 12: **il**, 13: **iu**, 14: **abstol**, 15: **m**, 16: **w**, 17: **z**, 18: **ldz**, 19: **work**, 20: **lwork**, 21: **iwork**, 22: **jfail**, 23: **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** $= i$, nag_lapack_dsyevx (f08fb) failed to converge; i eigenvectors failed to converge. Their indices are stored in array **jfail**.

info $>$ **n**

nag_lapack_dpotrf (f07fd) 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.

The complex analogue of this function is nag_lapack_zhegvx (f08sp).

9 Example

This example finds the eigenvalues in the half-open interval $(-1.0, 1.0]$, and corresponding eigenvectors, of the generalized symmetric eigenproblem $Az = \lambda Bz$, where

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

The example program for nag_lapack_dsygvd (f08sc) illustrates solving a generalized symmetric eigenproblem of the form $ABz = \lambda z$.

9.1 Program Text

```
     function f08sb_example

fprintf('f08sb example results\n\n');

% Upper triangular parts of symmetric matrix A and symmetric definite matrix B
uplo = 'Upper';
n = 4;
a = [0.24,  0.39,  0.42, -0.16;
     0,    -0.11,  0.79,  0.63;
     0,     0,    -0.25,  0.48;
     0,     0,     0,    -0.03];
b = [4.16, -3.12,  0.56, -0.10;
     0,     5.03, -0.83,  1.09;
     0,     0,     0.76,  0.34;
     0,     0,     0,     1.18];

% Generalized eigenvalues and eigenvectors for problem Az = lambda Bz
% Selecting eigenvalues in the range [-1,1]
itype = nag_int(1);
jobz  = 'Vectors';
range = 'Values in range';
[vl,vu] = deal(-1, 1);
[il,iu] = deal(nag_int(0), nag_int(0));
abstol = 0;
[~, ~, m, w, z, jfail, info] = ...
  f08sb( ...
```

```
  itype, jobz, range, uplo, a, b, vl, vu, il, iu, abstol);

% Normalize eigenvectors: largest element positive
for j = 1:m
  [~,k] = max(abs(z(:,j)));
  if z(k,j) < 0;
    z(:,j) = -z(:,j);
  end
end

disp('Selected Eigenvalues');
disp(w(1:m)');
disp('Corresponding eigenvectors');
disp(z);
```

9.2 Program Results

```
     f08sb example results

Selected Eigenvalues
   -0.4548    0.1001

Corresponding eigenvectors
   -0.3080   -0.4469
   -0.5329   -0.0371
    0.3496    0.0505
    0.6211    0.4743
```