

NAG Toolbox

nag_lapack_dspgvx (f08tb)

1 Purpose

nag_lapack_dspgvx (f08tb) 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, stored in packed storage, 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

```
[ap, bp, m, w, z, jfail, info] = nag_lapack_dspgvx(itype, jobz, range, uplo, n,
ap, bp, vl, vu, il, iu, abstol)
```

```
[ap, bp, m, w, z, jfail, info] = f08tb(itype, jobz, range, uplo, n, ap, bp, vl,
vu, il, iu, abstol)
```

3 Description

nag_lapack_dspgvx (f08tb) first performs a Cholesky factorization of the matrix B as $B = U^T U$, when **uplo** = 'U' or $B = LL^T$, 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

$$Z^T A Z = \Lambda \quad \text{and} \quad Z^T B Z = I,$$

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

$$Z^{-1} A Z^{-T} = \Lambda \quad \text{and} \quad Z^T B Z = I,$$

and for $BAz = \lambda z$ we have

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

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 **ilth** to **iuth** 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: **n** – INTEGER

n , the order of the matrices A and B .

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

6: **ap**(:) – REAL (KIND=nag_wp) array

The dimension of the array **ap** must be at least $\max(1, \mathbf{n} \times (\mathbf{n} + 1)/2)$

The upper or lower triangle of the n by n symmetric matrix A , packed by columns.

More precisely,

if **uplo** = 'U', the upper triangle of A must be stored with element A_{ij} in **ap**($i + j(j - 1)/2$) for $i \leq j$;

if **uplo** = 'L', the lower triangle of A must be stored with element A_{ij} in **ap**($i + (2n - j)(j - 1)/2$) for $i \geq j$.

7: **bp**(:) – REAL (KIND=nag_wp) array

The dimension of the array **bp** must be at least $\max(1, \mathbf{n} \times (\mathbf{n} + 1)/2)$

The upper or lower triangle of the n by n symmetric matrix B , packed by columns.

More precisely,

if **uplo** = 'U', the upper triangle of B must be stored with element B_{ij} in **bp**($i + j(j - 1)/2$) for $i \leq j$;

if **uplo** = 'L', the lower triangle of B must be stored with element B_{ij} in **bp**($i + (2n - j)(j - 1)/2$) for $i \geq j$.

8: **vl** – REAL (KIND=nag_wp)

9: **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 'T', **vl** and **vu** are not referenced.

Constraint: if **range** = 'V', **vl** < **vu**.

10: **il** – INTEGER

11: **iu** – INTEGER

If **range** = 'T', 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** = 'T' and $\mathbf{n} = 0$, **il** = 1 and **iu** = 0;

if **range** = 'T' and $\mathbf{n} > 0$, $1 \leq \mathbf{il} \leq \mathbf{iu} \leq \mathbf{n}$.

12: **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 \text{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 \text{x02am}(\)$. See Demmel and Kahan (1990).

5.2 Optional Input Parameters

None.

5.3 Output Parameters

1: **ap**(:) – REAL (KIND=nag_wp) array

The dimension of the array **ap** will be $\max(1, \mathbf{n} \times (\mathbf{n} + 1)/2)$

The contents of **ap** are destroyed.

- 2: **bp**(:) – REAL (KIND=nag_wp) array
 The dimension of the array **bp** will be $\max(1, \mathbf{n} \times (\mathbf{n} + 1)/2)$
 The triangular factor U or L from the Cholesky factorization $B = U^T U$ or $B = LL^T$, in the same storage format as B .
- 3: **m** – INTEGER
 The total number of eigenvalues found. $0 \leq \mathbf{m} \leq \mathbf{n}$.
 If **range** = 'A', **m** = **n**.
 If **range** = 'I', **m** = **iu** – **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 i th column of Z holding the eigenvector associated with **w**(i). The eigenvectors are normalized as follows:
 if **itype** = 1 or 2, $Z^T B Z = I$;
 if **itype** = 3, $Z^T B^{-1} Z = I$;
 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: **ap**, 7: **bp**, 8: **vl**, 9: **vu**, 10: **il**, 11: **iu**, 12: **abstol**, 13: **m**, 14: **w**, 15: **z**, 16: **ldz**, 17: **work**, 18: **iwork**, 19: **jfail**, 20: **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** (*warning*)

If **info** = i , nag_lapack_dspevx (f08gb) failed to converge; i eigenvectors failed to converge. Their indices are stored in array **jfail**.

info > **n**

nag_lapack_dpptrf (f07gd) returned an error code; i.e., if **info** = $\mathbf{n} + i$, for $1 \leq i \leq \mathbf{n}$, 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_zhpgvx (f08tp).

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_dspgvd (f08tc) illustrates solving a generalized symmetric eigenproblem of the form $ABz = \lambda z$.

9.1 Program Text

```
function f08tb_example

fprintf('f08tb example results\n\n');

% Symmetric matrices A and B stored in packed (Upper) format
n = nag_int(4);
uplo = 'U';
ap = [0.24;
      0.39; -0.11;
      0.42; 0.79; -0.25;
      -0.16; 0.63; 0.48; -0.03];
bp = [4.16;
      -3.12; 5.03;
      0.56; -0.83; 0.76;
      -0.10; 1.09; 0.34; 1.18];

% Eigenvalues in the range [-1,1] and corresponding eigenvectors
itype = nag_int(1);
jobz = 'Vectors';
range = 'Values in range';
vl = -1;      vu = 1;
```

```
il = nag_int(0); iu = nag_int(0);
abstol = 0;
[~, ~, m, w, z, jfail, info] = ...
f08tb( ...
    itype, jobz, range, uplo, n, ap, bp, vl, vu, il, iu, abstol);

disp('Selected Eigenvalues');
disp(w(1:m)');
% Normalize eigenvectors (wrt B): largest element positive
for j = 1:m
    [~,k] = max(abs(z(:,j)));
    if z(k,j) < 0;
        z(:,j) = -z(:,j);
    end
end
disp('Corresponding Eigenvectors');
disp(z);
```

9.2 Program Results

f08tb 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
```
