NAG Toolbox

nag_lapack_dspgvd (f08tc)

1 Purpose

nag_lapack_dspgvd (f08tc) computes all the eigenvalues and, optionally, the 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 format, and B is also positive definite. If eigenvectors are desired, it uses a divide-and-conquer algorithm.

2 Syntax

```
[ap, bp, w, z, info] = nag_lapack_dspgvd(itype, jobz, uplo, n, ap, bp)
[ap, bp, w, z, info] = f08tc(itype, jobz, uplo, n, ap, bp)
```

3 Description

nag_lapack_dspgvd (f08tc) 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 eigenvalues and, optionally, the 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

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: **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'.

4: **n** – INTEGER

n, the order of the matrices A and B.

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

5: **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 $\mathbf{ap}(i + j(j-1)/2)$ for $i \leq j$;

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

6: **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 $i \le j$;

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

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 $B = U^{\mathrm{T}}U$ or $B = LL^{\mathrm{T}}$, in the same storage format as B.

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

The eigenvalues in ascending order.

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

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

The second dimension of the array **z** will be $\max(1, \mathbf{n})$ if **jobz** = 'V' and 1 otherwise.

If **jobz** = 'V', **z** contains the matrix Z of eigenvectors. 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 **jobz** = 'N', **z** is not referenced.

5: **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: **uplo**, 4: **n**, 5: **ap**, 6: **bp**, 7: **w**, 8: **z**, 9: **ldz**, 10: **work**, 11: **lwork**, 12: **iwork**, 13: **liwork**, 14: **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 > 0

nag_lapack_dpptrf (f07gd) or nag_lapack_dspevd (f08gc) returned an error code:

$\leq$ **n** if **info** $= i$, nag_lapack_dspevd (f08gc) failed to converge; i off-diagonal elements of an intermediate tridiagonal form did not converge to zero;

$>$ **n** 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.

The example program below illustrates the computation of approximate 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_zhpgvd (f08tq).

9 Example

This example finds all the eigenvalues and eigenvectors of the generalized symmetric eigenproblem $ABz = \lambda z$, where

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

together with an estimate of the condition number of B, and approximate error bounds for the computed eigenvalues and eigenvectors.

The example program for nag_lapack_dspgv (f08ta) illustrates solving a generalized symmetric eigenproblem of the form $Az = \lambda Bz$.

9.1 Program Text

```
     function f08tc_example

fprintf('f08tc 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];

% Generalized eigenvalues for ABz = lambda z
itype = nag_int(2);
jobz = 'No vectors';
[~, L, w, z, info] = f08tc( ...
                            itype, jobz, uplo, n, ap, bp);

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

```
% Calculate condition number and error estimates
% Unpack ap and bp to calculate 1 norms of A and B
[a, ap, ifail] = f01za( ...
                        'Unpack', uplo, 'N', zeros(n,n), ap);
[b, bp, ifail] = f01za( ...
                        'Unpack', uplo, 'N', zeros(n,n), bp);
anorm = norm(a + a' - diag(diag(a)),1);
bnorm = norm(b + b' - diag(diag(b)),1);

% Estimate condition number
[rcond, info] = f07ug( ...
                       '1', uplo, 'N', n, L);

rcondb = rcond^2;
fprintf('Estimate of reciprocal condition number for B\n%12.1e\n', rcondb);

% Error bounds
t1 = x02aj/rcondb;
t2 = anorm*bnorm;
errbnd(1:n) = x02aj*t2 + t1.*abs(w);

fprintf('\n');
disp('Error estimate for the eigenvalues');
fprintf('%12.1e',errbnd);
fprintf('\n');
```

9.2 Program Results

```
     f08tc example results

Eigenvalues
   -3.5411   -0.3347    0.2983    2.2544

Estimate of reciprocal condition number for B
     5.8e-03

Error estimate for the eigenvalues
     7.0e-14     8.6e-15     7.9e-15     4.6e-14
```