NAG Toolbox

nag_lapack_zhpgvd (f08tq)

1 Purpose

nag_lapack_zhpgvd (f08tq) computes all the eigenvalues and, optionally, the eigenvectors of a complex generalized Hermitian-definite eigenproblem, of the form

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

where A and B are Hermitian, 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_zhpgvd(itype, jobz, uplo, n, ap, bp)
[ap, bp, w, z, info] = f08tq(itype, jobz, uplo, n, ap, bp)
```

3 Description

nag_lapack_zhpgvd (f08tq) first performs a Cholesky factorization of the matrix B as $B = U^{\mathrm{H}}U$, when **uplo** $=$ 'U' or $B = LL^{\mathrm{H}}$, 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

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

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

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

and for $BAz = \lambda z$ we have

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

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: **n** ≥ 0.

5: **ap**(:) – COMPLEX (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 Hermitian 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 $i \geq j$.

6: **bp**(:) – COMPLEX (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 Hermitian 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 $i \geq j$.

5.2 Optional Input Parameters

None.

5.3 Output Parameters

1: **ap**(:) – COMPLEX (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**(:) – COMPLEX (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 $B = U^{\mathrm{H}}U$ or $B = LL^{\mathrm{H}}$, in the same storage format as B.

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

The eigenvalues in ascending order.

4: **z**(ldz, :) – COMPLEX (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 $\max(1, \mathbf{n})$ 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, $Z^{\mathrm{H}}BZ = I$;

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

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: **rwork**, 13: **lrwork**, 14: **iwork**, 15: **liwork**, 16: **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_zpptrf (f07gr) or nag_lapack_zhpevd (f08gq) returned an error code:

$\leq$ **n** if **info** $= i$, nag_lapack_zhpevd (f08gq) 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 real analogue of this function is nag_lapack_dspgvd (f08tc).

9 Example

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

$$A = \begin{pmatrix} -7.36 & 0.77 - 0.43i & -0.64 - 0.92i & 3.01 - 6.97i \\ 0.77 + 0.43i & 3.49 & 2.19 + 4.45i & 1.90 + 3.73i \\ -0.64 + 0.92i & 2.19 - 4.45i & 0.12 & 2.88 - 3.17i \\ 3.01 + 6.97i & 1.90 - 3.73i & 2.88 + 3.17i & -2.54 \end{pmatrix}$$

and

$$B = \begin{pmatrix} 3.23 & 1.51 - 1.92i & 1.90 + 0.84i & 0.42 + 2.50i \\ 1.51 + 1.92i & 3.58 & -0.23 + 1.11i & -1.18 + 1.37i \\ 1.90 - 0.84i & -0.23 - 1.11i & 4.09 & 2.33 - 0.14i \\ 0.42 - 2.50i & -1.18 - 1.37i & 2.33 + 0.14i & 4.29 \end{pmatrix},$$

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_zhpgv (f08tn) illustrates solving a generalized Hermitian eigenproblem of the form $Az = \lambda Bz$.

9.1 Program Text

```
     function f08tq_example

fprintf('f08tq example results\n\n');

% Hermitian matrices A and B stored in packed (Upper) format
n = nag_int(4);
uplo = 'U';
ap = [-7.36;
        0.77 - 0.43i;  3.49 + 0i;
       -0.64 - 0.92i;  2.19 + 4.45i;  0.12 + 0i;
        3.01 - 6.97i;  1.90 + 3.73i;  2.88 - 3.17i; -2.54 + 0i];
bp = [ 3.23;
        1.51 - 1.92i;  3.58 + 0i;
        1.90 + 0.84i; -0.23 + 1.11i;  4.09 + 0i;
```

```
       0.42 + 2.50i; -1.18 + 1.37i;  2.33 - 0.14i;  4.29 + 0i];

% Eigenvalues only for AB z = lmbda z
itype = nag_int(2);
jobz = 'No vectors';
[~, ~, w, ~, info] = f08tq( ...
                            itype, jobz, uplo, n, ap, bp);

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

9.2 Program Results

```
     f08tq example results

Eigenvalues
  -61.7321   -6.6195    0.0725   43.1883
```