NAG Toolbox

nag_lapack_zhpgvx (f08tp)

1 Purpose

nag_lapack_zhpgvx (f08tp) computes selected eigenvalues and, optionally, 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. 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_zhpgvx(itype, jobz, range, uplo, n, ap, bp, vl, vu, il, iu, abstol)
[ap, bp, m, w, z, jfail, info] = f08tp(itype, jobz, range, uplo, n, ap, bp, vl, vu, il, iu, abstol)
```

3 Description

nag_lapack_zhpgvx (f08tp) 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 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

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

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

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: **n** – INTEGER

n, the order of the matrices A and B.

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

6: **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 $i \geq j$.

7: **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 $i \geq j$.

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 'I', **vl** and **vu** are not referenced.

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

10: **il** – INTEGER

11: **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**.

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

$$\textbf{abstol} + \epsilon \max(|a|, |b|),$$

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

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: **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, :) – 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 $\max(1, \mathbf{m})$ 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 $\mathbf{w}(i)$. 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 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: **rwork**, 19: **iwork**, 20: **jfail**, 21: **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_zhpevx (f08gp) failed to converge; i eigenvectors failed to converge. Their indices are stored in array **jfail**.

info $>$ **n**

nag_lapack_zpptrf (f07gr) 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 real analogue of this function is nag_lapack_dspgvx (f08tb).

9 Example

This example finds the eigenvalues in the half-open interval $(-3, 3]$, and corresponding eigenvectors, of the generalized Hermitian eigenproblem $Az = \lambda Bz$, 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

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

The example program for nag_lapack_zhpgvd (f08tq) illustrates solving a generalized symmetric eigenproblem of the form $ABz = \lambda z$.

9.1 Program Text

```
function f08tp_example

fprintf('f08tp 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 in range [-3,3] and corresponding eigenvectors for Az=lambda Bz
itype = nag_int(1);
jobz  = 'Vectors';
range = 'Values in range';
vl = -3;         vu = 3;
il = nag_int(0); iu = il;
abstol = 0;
[~, ~, m, w, z, jfail, info] = ...
f08tp( ...
       itype, jobz, range, uplo, n, ap, bp, vl, vu, il, iu, abstol);

disp('Selected Eigenvalues');
disp(w(1:m)');

% Normalize eigenvectors: largest elements are real
for i = 1:m
  [~,k] = max(abs(real(z(:,i)))+abs(imag(z(:,i))));
  z(:,i) = z(:,i)*conj(z(k,i))/abs(z(k,i));
end
disp('Corresponding Eigenvectors');
disp(z);
```

9.2 Program Results

```
     f08tp example results

Selected Eigenvalues
   -2.9936    0.5047

Corresponding Eigenvectors
  -0.6626 + 0.2258i   0.6462 + 0.0000i
  -0.1164 - 0.0178i   0.1216 - 0.4788i
   0.9098 + 0.0000i  -0.1344 - 0.3059i
  -0.6120 - 0.5348i   0.2924 + 0.5987i
```