# NAG Toolbox

# nag_lapack_zhpgst (f08ts)

## 1 Purpose

nag_lapack_zhpgst (f08ts) reduces a complex Hermitian-definite generalized eigenproblem $Az = \lambda Bz$, $ABz = \lambda z$ or $BAz = \lambda z$ to the standard form $Cy = \lambda y$, where $A$ is a complex Hermitian matrix and $B$ has been factorized by nag_lapack_zpptrf (f07gr), using packed storage.

## 2 Syntax

```
[ap, info] = nag_lapack_zhpgst(itype, uplo, n, ap, bp)
[ap, info] = f08ts(itype, uplo, n, ap, bp)
```

## 3 Description

To reduce the complex Hermitian-definite generalized eigenproblem $Az = \lambda Bz$, $ABz = \lambda z$ or $BAz = \lambda z$ to the standard form $Cy = \lambda y$ using packed storage, nag_lapack_zhpgst (f08ts) must be preceded by a call to nag_lapack_zpptrf (f07gr) which computes the Cholesky factorization of $B$; $B$ must be positive definite.

The different problem types are specified by the argument **itype**, as indicated in the table below. The table shows how $C$ is computed by the function, and also how the eigenvectors $z$ of the original problem can be recovered from the eigenvectors of the standard form.

| **itype** | Problem | **uplo** | $B$ | $C$ | $z$ |
|---|---|---|---|---|---|
| 1 | $Az = \lambda Bz$ | 'U'<br>'L' | $U^{\mathrm{H}}U$<br>$LL^{\mathrm{H}}$ | $U^{-\mathrm{H}}AU^{-1}$<br>$L^{-1}AL^{-\mathrm{H}}$ | $U^{-1}y$<br>$L^{-\mathrm{H}}y$ |
| 2 | $ABz = \lambda z$ | 'U'<br>'L' | $U^{\mathrm{H}}U$<br>$LL^{\mathrm{H}}$ | $UAU^{\mathrm{H}}$<br>$L^{\mathrm{H}}AL$ | $U^{-1}y$<br>$L^{-\mathrm{H}}y$ |
| 3 | $BAz = \lambda z$ | 'U'<br>'L' | $U^{\mathrm{H}}U$<br>$LL^{\mathrm{H}}$ | $UAU^{\mathrm{H}}$<br>$L^{\mathrm{H}}AL$ | $U^{\mathrm{H}}y$<br>$Ly$ |

## 4 References

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

Indicates how the standard form is computed.

**itype** $= 1$

if **uplo** $=$ 'U', $C = U^{-\mathrm{H}}AU^{-1}$;

if **uplo** $=$ 'L', $C = L^{-1}AL^{-\mathrm{H}}$.

**itype** $= 2$ or $3$

if **uplo** = 'U', $C = UAU^{\mathrm{H}}$;

if **uplo** = 'L', $C = L^{\mathrm{H}}AL$.

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

2: **uplo** – CHARACTER(1)

Indicates whether the upper or lower triangular part of $A$ is stored and how $B$ has been factorized.

**uplo** = 'U'
The upper triangular part of $A$ is stored and $B = U^{\mathrm{H}}U$.

**uplo** = 'L'
The lower triangular part of $A$ is stored and $B = LL^{\mathrm{H}}$.

*Constraint*: **uplo** = 'U' or 'L'.

3: **n** – INTEGER

$n$, the order of the matrices $A$ and $B$.

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

4: **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 $\mathbf{ap}(i + j(j - 1)/2)$ 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 $i \geq j$.

5: **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 Cholesky factor of $B$ as specified by **uplo** and returned by nag_lapack_zpptrf (f07gr).

## 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 upper or lower triangle of **ap** stores the corresponding upper or lower triangle of $C$ as specified by **itype** and **uplo**, using the same packed storage format as described above.

2: **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: **uplo**, 3: **n**, 4: **ap**, 5: **bp**, 6: **info**.

## 7 Accuracy

Forming the reduced matrix $C$ is a stable procedure. However it involves implicit multiplication by $B^{-1}$ if (**itype** $= 1$) or $B$ (if **itype** $= 2$ or 3). When nag_lapack_zhpgst (f08ts) is used as a step in the computation of eigenvalues and eigenvectors of the original problem, there may be a significant loss of accuracy if $B$ is ill-conditioned with respect to inversion. See the document for nag_lapack_zhegv (f08sn) for further details.

## 8 Further Comments

The total number of real floating-point operations is approximately $4n^3$.

The real analogue of this function is nag_lapack_dspgst (f08te).

## 9 Example

This example computes all the eigenvalues of $Az = \lambda Bz$, where

$$A = \begin{pmatrix} -7.36 + 0.00i & 0.77 - 0.43i & -0.64 - 0.92i & 3.01 - 6.97i \\ 0.77 + 0.43i & 3.49 + 0.00i & 2.19 + 4.45i & 1.90 + 3.73i \\ -0.64 + 0.92i & 2.19 - 4.45i & 0.12 + 0.00i & 2.88 - 3.17i \\ 3.01 + 6.97i & 1.90 - 3.73i & 2.88 + 3.17i & -2.54 + 0.00i \end{pmatrix}$$

and

$$B = \begin{pmatrix} 3.23 + 0.00i & 1.51 - 1.92i & 1.90 + 0.84i & 0.42 + 2.50i \\ 1.51 + 1.92i & 3.58 + 0.00i & -0.23 + 1.11i & -1.18 + 1.37i \\ 1.90 - 0.84i & -0.23 - 1.11i & 4.09 + 0.00i & 2.33 - 0.14i \\ 0.42 - 2.50i & -1.18 - 1.37i & 2.33 + 0.14i & 4.29 + 0.00i \end{pmatrix},$$

using packed storage. Here $B$ is Hermitian positive definite and must first be factorized by nag_lapack_zpptrf (f07gr). The program calls nag_lapack_zhpgst (f08ts) to reduce the problem to the standard form $Cy = \lambda y$; then nag_lapack_zhptrd (f08gs) to reduce $C$ to tridiagonal form, and nag_lapack_dsterf (f08jf) to compute the eigenvalues.

### 9.1 Program Text

```
     function f08ts_example

fprintf('f08ts example results\n\n');

% Hermitian matrices A and B stored in packed (Lower) format
uplo = 'L';
n = nag_int(4);
ap = [-7.36;  0.77 + 0.43i; -0.64 + 0.92i;  3.01 + 6.97i;
              3.49 + 0i;     2.19 - 4.45i;  1.90 - 3.73i;
                             0.12 + 0i;     2.88 + 3.17i;
                                           -2.54 + 0i];
bp = [ 3.23;  1.51 + 1.92i;  1.90 - 0.84i;  0.42 - 2.50i;
              3.58 + 0i;    -0.23 - 1.11i; -1.18 - 1.37i;
                             4.09 - 0i;     2.33 + 0.14i;
                                            4.29 - 0i];

% Compute Cholesky factorization of B
[lp, info] = f07gr( ...
     uplo, n, bp);
```

```
% Reduce problem to standard form Cy = lambda y,
% where y = L^H x, C = L^-1 A L^-H
itype = nag_int(1);
[cp, info] = f08ts( ...
    itype, uplo, n, ap, lp);

% Reduce C to tridiagonal form T = Q^H C Q
[qp, d, e, tau, info] = f08gs( ...
        uplo, n, cp);

% Calculate eigenvalues of T (same as C)
[w, ~, info] = f08jf(d, e);

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

### 9.2 Program Results

```
     f08ts example results

Eigenvalues
   -5.9990   -2.9936    0.5047    3.9990
```