NAG Toolbox

nag_lapack_zhbgst (f08us)

1 Purpose

nag_lapack_zhbgst (f08us) reduces a complex Hermitian-definite generalized eigenproblem $Az = \lambda Bz$ to the standard form $Cy = \lambda y$, where A and B are band matrices, A is a complex Hermitian matrix, and B has been factorized by nag_lapack_zpbstf (f08ut).

2 Syntax

```
[ab, x, info] = nag_lapack_zhbgst(vect, uplo, ka, kb, ab, bb, 'n', n)
[ab, x, info] = f08us(vect, uplo, ka, kb, ab, bb, 'n', n)
```

3 Description

To reduce the complex Hermitian-definite generalized eigenproblem $Az = \lambda Bz$ to the standard form $Cy = \lambda y$, where A, B and C are banded, nag_lapack_zhbgst (f08us) must be preceded by a call to nag_lapack_zpbstf (f08ut) which computes the split Cholesky factorization of the positive definite matrix B: $B = S^{\mathrm{H}}S$. The split Cholesky factorization, compared with the ordinary Cholesky factorization, allows the work to be approximately halved.

This function overwrites A with $C = X^{\mathrm{H}}AX$, where $X = S^{-1}Q$ and Q is a unitary matrix chosen (implicitly) to preserve the bandwidth of A. The function also has an option to allow the accumulation of X, and then, if z is an eigenvector of C, Xz is an eigenvector of the original system.

4 References

Crawford C R (1973) Reduction of a band-symmetric generalized eigenvalue problem *Comm. ACM* **16** 41–44

Kaufman L (1984) Banded eigenvalue solvers on vector machines *ACM Trans. Math. Software* **10** 73–86

5 Parameters

5.1 Compulsory Input Parameters

1: **vect** – CHARACTER(1)

Indicates whether X is to be returned.

vect = 'N'
X is not returned.

vect = 'V'
X is returned.

Constraint: **vect** = 'N' or 'V'.

2: **uplo** – CHARACTER(1)

Indicates whether the upper or lower triangular part of A is stored.

uplo = 'U'
The upper triangular part of A is stored.

uplo = 'L'

The lower triangular part of A is stored.

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

3: **ka** – INTEGER

If **uplo** = 'U', the number of superdiagonals, k_a, of the matrix A.

If **uplo** = 'L', the number of subdiagonals, k_a, of the matrix A.

Constraint: **ka** ≥ 0.

4: **kb** – INTEGER

If **uplo** = 'U', the number of superdiagonals, k_b, of the matrix B.

If **uplo** = 'L', the number of subdiagonals, k_b, of the matrix B.

Constraint: **ka** $\geq$ **kb** ≥ 0.

5: **ab**$(ldab,:)$ – COMPLEX (KIND=nag_wp) array

The first dimension of the array **ab** must be at least **ka** $+ 1$.

The second dimension of the array **ab** must be at least $\max(1,$ **n**$)$.

The upper or lower triangle of the n by n Hermitian band matrix A.

The matrix is stored in rows 1 to $k_a + 1$, more precisely,

if **uplo** = 'U', the elements of the upper triangle of A within the band must be stored with element A_{ij} in **ab**$(k_a + 1 + i - j, j)$ for $\max(1, j - k_a) \leq i \leq j$;

if **uplo** = 'L', the elements of the lower triangle of A within the band must be stored with element A_{ij} in **ab**$(1 + i - j, j)$ for $j \leq i \leq \min(n, j + k_a)$.

6: **bb**$(ldbb,:)$ – COMPLEX (KIND=nag_wp) array

The first dimension of the array **bb** must be at least **kb** $+ 1$.

The second dimension of the array **bb** must be at least $\max(1,$ **n**$)$.

The banded split Cholesky factor of B as specified by **uplo**, **n** and **kb** and returned by nag_lapack_zpbstf (f08ut).

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the second dimension of the arrays **ab**, **bb**.

n, the order of the matrices A and B.

Constraint: **n** ≥ 0.

5.3 Output Parameters

1: **ab**$(ldab,:)$ – COMPLEX (KIND=nag_wp) array

The first dimension of the array **ab** will be **ka** $+ 1$.

The second dimension of the array **ab** will be $\max(1,$ **n**$)$.

The upper or lower triangle of **ab** stores the corresponding upper or lower triangle of C as specified by **uplo**.

2: **x**(ldx, :) – COMPLEX (KIND=nag_wp) array

The first dimension, ldx, of the array **x** will be

if **vect** = 'V', $ldx = \max(1, \mathbf{n})$;
if **vect** = 'N', $ldx = 1$.

The second dimension of the array **x** will be $\max(1, \mathbf{n})$ if **vect** = 'V' and at least 1 if **vect** = 'N'.

The n by n matrix $X = S^{-1}Q$, if **vect** = 'V'.

If **vect** = 'N', **x** is not referenced.

3: **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: **vect**, 2: **uplo**, 3: **n**, 4: **ka**, 5: **kb**, 6: **ab**, 7: **ldab**, 8: **bb**, 9: **ldbb**, 10: **x**, 11: **ldx**, 12: **work**, 13: **rwork**, 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.

7 Accuracy

Forming the reduced matrix C is a stable procedure. However it involves implicit multiplication by B^{-1}. When nag_lapack_zhbgst (f08us) 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.

8 Further Comments

The total number of real floating-point operations is approximately $20n^2k_B$, when **vect** = 'N', assuming $n \gg k_A, k_B$; there are an additional $5n^3(k_B/k_A)$ operations when **vect** = 'V'.

The real analogue of this function is nag_lapack_dsbgst (f08ue).

9 Example

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

$$A = \begin{pmatrix} -1.13 + 0.00i & 1.94 - 2.10i & -1.40 + 0.25i & 0.00 + 0.00i \\ 1.94 + 2.10i & -1.91 + 0.00i & -0.82 - 0.89i & -0.67 + 0.34i \\ -1.40 - 0.25i & -0.82 + 0.89i & -1.87 + 0.00i & -1.10 - 0.16i \\ 0.00 + 0.00i & -0.67 - 0.34i & -1.10 + 0.16i & 0.50 + 0.00i \end{pmatrix}$$

and

$$B = \begin{pmatrix} 9.89 + 0.00i & 1.08 - 1.73i & 0.00 + 0.00i & 0.00 + 0.00i \\ 1.08 + 1.73i & 1.69 + 0.00i & -0.04 + 0.29i & 0.00 + 0.00i \\ 0.00 + 0.00i & -0.04 - 0.29i & 2.65 + 0.00i & -0.33 + 2.24i \\ 0.00 + 0.00i & 0.00 + 0.00i & -0.33 - 2.24i & 2.17 + 0.00i \end{pmatrix}.$$

Here A is Hermitian, B is Hermitian positive definite, and A and B are treated as band matrices. B must first be factorized by nag_lapack_zpbstf (f08ut). The program calls nag_lapack_zhbgst (f08us) to reduce the problem to the standard form $Cy = \lambda y$, then nag_lapack_zhbtrd (f08hs) to reduce C to tridiagonal form, and nag_lapack_dsterf (f08jf) to compute the eigenvalues.

9.1 Program Text

```
     function f08us_example

fprintf('f08us example results\n\n');

% Sove Az = lambda Bz
% A and B are the Hermitian banded positive definite matrices:
n = 4;
ka = nag_int(2);
a = [ -1.13+0.00i  1.94-2.10i -1.40+0.25i  0.00+0.00i;
       1.94+2.10i -1.91+0.00i -0.82-0.89i -0.67+0.34i;
      -1.40-0.25i -0.82+0.89i -1.87+0.00i -1.10-0.16i;
       0.00+0.00i -0.67-0.34i -1.10+0.16i  0.50+0.00i];

kb = nag_int(1);
b = [  9.89+0.00i  1.08-1.73i  0.00+0.00i  0.00+0.00i;
       1.08+1.73i  1.69+0.00i -0.04+0.29i  0.00+0.00i;
       0.00+0.00i -0.04-0.29i  2.65+0.00i -0.33+2.24i;
       0.00+0.00i  0.00+0.00i -0.33-2.24i  2.17+0.00i];

% Convert to general banded format ...
[~, ab, ifail] = f01zd( ...
 'P', ka, ka, a, complex(zeros(ka+ka+1,n)));
[~, bb, ifail] = f01zd(...
        'P', kb, kb, b, complex(zeros(kb+kb+1,n)));
% ... and chop to give 'Upper' Hermitian banded format
ab = ab(1:ka+1,1:n);
bb = bb(1:kb+1,1:n);

% Factorize B
uplo = 'Upper';
[ub, info] = f08ut( ...
     uplo, kb, bb);

% Reduce problem to standard form Cy = lambda*y
vect = 'N';
[cb, x, info] = f08us( ...
        vect, uplo, ka, kb, ab, ub);

% Find eigenvalues lambda
jobz = 'No Vectors';
[~, w, ~, info] = f08hn( ...
  jobz, uplo, ka, cb);

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

9.2 Program Results

```
     f08us example results

Eigenvalues:
   -6.6089   -2.0416    0.1603    1.7712
```