NAG Library Routine Document

F08UEF (DSBGST)

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

F08UEF (DSBGST) reduces a real symmetric-definite generalized eigenproblem $Az = \lambda Bz$ to the standard form $Cy = \lambda y$, where A and B are band matrices, A is a real symmetric matrix, and B has been factorized by F08UFF (DPBSTF).

2 Specification

```
      SUBROUTINE F08UEF(VECT, UPLO, N, KA, KB, AB, LDAB, BB, LDBB, X, LDX,
     1                  WORK, INFO)
      INTEGER           N, KA, KB, LDAB, LDBB, LDX, INFO
      double precision  AB(LDAB,*), BB(LDBB,*), X(LDX,*), WORK(2*N)
      CHARACTER*1       VECT, UPLO
```

The routine may be called by its LAPACK name ***dsbgst***.

3 Description

To reduce the real symmetric-definite generalized eigenproblem $Az = \lambda Bz$ to the standard form $Cy = \lambda y$, where A, B and C are banded, F08UEF (DSBGST) must be preceded by a call to F08UFF (DPBSTF) which computes the split Cholesky factorization of the positive-definite matrix B: $B = S^{\mathrm{T}}S$. The split Cholesky factorization, compared with the ordinary Cholesky factorization, allows the work to be approximately halved.

This routine overwrites A with $C = X^{\mathrm{T}}AX$, where $X = S^{-1}Q$ and Q is a orthogonal matrix chosen (implicitly) to preserve the bandwidth of A. The routine 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

1: VECT – CHARACTER*1 *Input*

On entry: 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 *Input*

On entry: 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: N – INTEGER *Input*

On entry: n, the order of the matrices A and B.

Constraint: $\mathrm{N} \geq 0$.

4: KA – INTEGER *Input*

On entry: 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: $\mathrm{KA} \geq 0$.

5: KB – INTEGER *Input*

On entry: 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: $\mathrm{KA} \geq \mathrm{KB} \geq 0$.

6: AB(LDAB,$*$) – ***double precision*** array *Input/Output*

Note: the second dimension of the array AB must be at least $\max(1,\mathrm{N})$.

On entry: the upper or lower triangle of the n by n symmetric 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 $\mathrm{AB}(k_a + 1 + i - j, j)$ 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 $\mathrm{AB}(1 + i - j, j)$ for $j \leq i \leq \min(n, j + k_a)$.

On exit: the upper or lower triangle of AB is overwritten by the corresponding upper or lower triangle of C as specified by UPLO.

7: LDAB – INTEGER *Input*

On entry: the first dimension of the array AB as declared in the (sub)program from which F08UEF (DSBGST) is called.

Constraint: $\mathrm{LDAB} \geq \mathrm{KA} + 1$.

8: BB(LDBB,$*$) – ***double precision*** array *Input*

Note: the second dimension of the array BB must be at least $\max(1,\mathrm{N})$.

On entry: the banded split Cholesky factor of B as specified by UPLO, N and KB and returned by F08UFF (DPBSTF).

9: LDBB – INTEGER *Input*

On entry: the first dimension of the array BB as declared in the (sub)program from which F08UEF (DSBGST) is called.

Constraint: $\mathrm{LDBB} \geq \mathrm{KB} + 1$.

10: X(LDX,$*$) – ***double precision*** array *Output*

Note: the second dimension of the array X must be at least $\max(1, \mathrm{N})$ if VECT = 'V' and at least 1 if VECT = 'N'.

On exit: the n by n matrix $X = S^{-1}Q$, if VECT = 'V'.

If VECT = 'N', X is not referenced.

11: LDX – INTEGER *Input*

On entry: the first dimension of the array X as declared in the (sub)program from which F08UEF (DSBGST) is called.

Constraints:

if VECT = 'V', LDX $\geq \max(1, \mathrm{N})$;
if VECT = 'N', LDX ≥ 1.

12: WORK($2 \times \mathrm{N}$) – ***double precision*** array *Workspace*

13: INFO – INTEGER *Output*

On exit: INFO = 0 unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

INFO < 0

If INFO $= -i$, argument i had an illegal value. An explanatory message is output, and execution of the program is terminated.

7 Accuracy

Forming the reduced matrix C is a stable procedure. However it involves implicit multiplication by B^{-1}. When F08UEF (DSBGST) 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 floating-point operations is approximately $6n^2k_B$, when VECT = 'N', assuming $n \gg k_A, k_B$; there are an additional $(3/2)n^3(k_B/k_A)$ operations when VECT = 'V'.

The complex analogue of this routine is F08USF (ZHBGST).

9 Example

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

$$A = \begin{pmatrix} 0.24 & 0.39 & 0.42 & 0.00 \\ 0.39 & -0.11 & 0.79 & 0.63 \\ 0.42 & 0.79 & -0.25 & 0.48 \\ 0.00 & 0.63 & 0.48 & -0.03 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 2.07 & 0.95 & 0.00 & 0.00 \\ 0.95 & 1.69 & -0.29 & 0.00 \\ 0.00 & -0.29 & 0.65 & -0.33 \\ 0.00 & 0.00 & -0.33 & 1.17 \end{pmatrix}.$$

Here A is symmetric, B is symmetric positive-definite, and A and B are treated as band matrices. B must first be factorized by F08UFF (DPBSTF). The program calls F08UEF (DSBGST) to reduce the problem to the standard form $Cy = \lambda y$, then F08HEF (DSBTRD) to reduce C to tridiagonal form, and F08JFF (DSTERF) to compute the eigenvalues.

9.1 Program Text

```
*     F08UEF Example Program Text
*     Mark 19 Release. NAG Copyright 1999.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, KMAX, LDAB, LDBB, LDX
      PARAMETER        (NMAX=8,KMAX=8,LDAB=KMAX-1,LDBB=KMAX-1,LDX=NMAX)
*     .. Local Scalars ..
      INTEGER          I, INFO, J, KA, KB, N
      CHARACTER        UPLO
*     .. Local Arrays ..
      DOUBLE PRECISION AB(LDAB,NMAX), BB(LDBB,NMAX), D(NMAX), E(NMAX-1),
     +                 WORK(2*NMAX), X(LDX,NMAX)
*     .. External Subroutines ..
      EXTERNAL         DPBSTF, DSBGST, DSBTRD, DSTERF
*     .. Intrinsic Functions ..
      INTRINSIC        MAX, MIN
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08UEF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N, KA, KB
      IF (N.LE.NMAX .AND. KA.LE.KMAX .AND. KB.LE.KA) THEN
*
*        Read A and B from data file
*
         READ (NIN,*) UPLO
         IF (UPLO.EQ.'U') THEN
            DO 20 I = 1, N
               READ (NIN,*) (AB(KA+1+I-J,J),J=I,MIN(N,I+KA))
   20       CONTINUE
            DO 40 I = 1, N
               READ (NIN,*) (BB(KB+1+I-J,J),J=I,MIN(N,I+KB))
   40       CONTINUE
         ELSE IF (UPLO.EQ.'L') THEN
            DO 60 I = 1, N
               READ (NIN,*) (AB(1+I-J,J),J=MAX(1,I-KA),I)
   60       CONTINUE
            DO 80 I = 1, N
               READ (NIN,*) (BB(1+I-J,J),J=MAX(1,I-KB),I)
   80       CONTINUE
         END IF
*
*        Compute the split Cholesky factorization of B
*
         CALL DPBSTF(UPLO,N,KB,BB,LDBB,INFO)
*
         WRITE (NOUT,*)
         IF (INFO.GT.0) THEN
            WRITE (NOUT,*) 'B is not positive-definite.'
         ELSE
*
*           Reduce the problem to standard form C*y = lambda*y, storing
*           the result in A
*
            CALL DSBGST('N',UPLO,N,KA,KB,AB,LDAB,BB,LDBB,X,LDX,WORK,
     +                  INFO)
*
*           Reduce C to tridiagonal form T = (Q**T)*C*Q
*
            CALL DSBTRD('N',UPLO,N,KA,AB,LDAB,D,E,X,LDX,WORK,INFO)
*
*           Calclate the eigenvalues of T (same as C)
*
            CALL DSTERF(N,D,E,INFO)
*
            IF (INFO.GT.0) THEN
               WRITE (NOUT,*) 'Failure to converge.'
            ELSE
```

```
*
*              Print eigenvalues
*
               WRITE (NOUT,*) 'Eigenvalues'
               WRITE (NOUT,99999) (D(I),I=1,N)
            END IF
         END IF
      END IF
*
99999 FORMAT (3X,(8F8.4))
      END
```

9.2 Program Data

```
F08UEF Example Program Data
  4  2  1                    :Values of N, KA and KB
  'L'                        :Value of UPLO
  0.24
  0.39  -0.11
  0.42   0.79  -0.25
         0.63   0.48  -0.03  :End of matrix A
  2.07
  0.95   1.69
        -0.29   0.65
               -0.33   1.17  :End of matrix B
```

9.3 Program Results

```
 F08UEF Example Program Results

 Eigenvalues
    -0.8305 -0.6401  0.0992  1.8525
```