NAG Library Routine Document

F08TBF (DSPGVX)

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

F08TBF (DSPGVX) computes selected eigenvalues and, optionally, eigenvectors of a real generalized symmetric-definite eigenproblem, of the form

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

where A and B are symmetric, stored in packed storage, 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 Specification

```
SUBROUTINE F08TBF (ITYPE, JOBZ, RANGE, UPLO, N, AP, BP, VL, VU, IL, IU,    &
                   ABSTOL, M, W, Z, LDZ, WORK, IWORK, JFAIL, INFO)

INTEGER            ITYPE, N, IL, IU, M, LDZ, IWORK(5*N), JFAIL(*), INFO
REAL (KIND=nag_wp) AP(*), BP(*), VL, VU, ABSTOL, W(N), Z(LDZ,*),           &
                   WORK(8*N)
CHARACTER(1)       JOBZ, RANGE, UPLO
```

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

3 Description

F08TBF (DSPGVX) first performs a Cholesky factorization of the matrix B as $B = U^{\mathrm{T}}U$, when UPLO = 'U' or $B = LL^{\mathrm{T}}$, 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

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

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

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

and for $BAz = \lambda z$ we have

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

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

1: ITYPE – INTEGER *Input*

On entry: 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) *Input*

On entry: 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) *Input*

On entry: if RANGE = 'A', all eigenvalues will be found.

If RANGE = 'V', all eigenvalues in the half-open interval $(\mathrm{VL}, \mathrm{VU}]$ will be found.

If RANGE = 'I', the ILth to IUth eigenvalues will be found.

Constraint: RANGE = 'A', 'V' or 'I'.

4: UPLO – CHARACTER(1) *Input*

On entry: 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 *Input*

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

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

6: AP($*$) – REAL (KIND=nag_wp) array *Input/Output*

Note: the dimension of the array AP must be at least $\max(1, N \times (N+1)/2)$.

On entry: the upper or lower triangle of the n by n symmetric 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 $i \geq j$.

On exit: the contents of AP are destroyed.

7: BP($*$) – REAL (KIND=nag_wp) array *Input/Output*

Note: the dimension of the array BP must be at least $\max(1, N \times (N+1)/2)$.

On entry: the upper or lower triangle of the n by n symmetric 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 $i \geq j$.

On exit: the triangular factor U or L from the Cholesky factorization $B = U^{T}U$ or $B = LL^{T}$, in the same storage format as B.

8: VL – REAL (KIND=nag_wp) *Input*
9: VU – REAL (KIND=nag_wp) *Input*

On entry: 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 *Input*
11: IU – INTEGER *Input*

On entry: 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) *Input*

On entry: 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

$$\text{ABSTOL} + \epsilon \max(|a|, |b|),$$

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 $2 \times$ X02AMF( ), not zero. If this routine returns with INFO = 1 to N, indicating that some eigenvectors did not converge, try setting ABSTOL to $2 \times$ X02AMF( ). See Demmel and Kahan (1990).

13: M – INTEGER *Output*

On exit: the total number of eigenvalues found. $0 \leq \text{M} \leq \text{N}$.

If RANGE = 'A', M = N.

If RANGE = 'I', M = IU − IL + 1.

14: W(N) – REAL (KIND=nag_wp) array *Output*

On exit: the first M elements contain the selected eigenvalues in ascending order.

15: Z(LDZ, ∗) – REAL (KIND=nag_wp) array *Output*

Note: the second dimension of the array Z must be at least max(1, M) if JOBZ = 'V', and at least 1 otherwise.

On exit: 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 W(i). The eigenvectors are normalized as follows:

if ITYPE = 1 or 2, $Z^{\mathrm{T}}BZ = I$;

if ITYPE = 3, $Z^{\mathrm{T}}B^{-1}Z = I$;

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.

Note: you must ensure that at least max(1, M) columns are supplied in the array Z; if RANGE = 'V', the exact value of M is not known in advance and an upper bound of at least N must be used.

16: LDZ – INTEGER *Input*

On entry: the first dimension of the array Z as declared in the (sub)program from which F08TBF (DSPGVX) is called.

Constraints:

if JOBZ = 'V', LDZ ≥ max(1, N);
otherwise LDZ ≥ 1.

17: WORK(8 × N) – REAL (KIND=nag_wp) array *Workspace*

18: IWORK(5 × N) – INTEGER array *Workspace*

19: JFAIL(∗) – INTEGER array *Output*

Note: the dimension of the array JFAIL must be at least max(1, N).

On exit: 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.

20: 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.

INFO $= 1$ to N

If INFO $= i$, F08GBF (DSPEVX) failed to converge; i eigenvectors failed to converge. Their indices are stored in array JFAIL.

INFO $>$ N

F07GDF (DPPTRF) returned an error code; i.e., if INFO $= \mathrm{N} + i$, for $1 \leq i \leq \mathrm{N}$, 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 Parallelism and Performance

F08TBF (DSPGVX) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F08TBF (DSPGVX) makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

The total number of floating-point operations is proportional to n^3.

The complex analogue of this routine is F08TPF (ZHPGVX).

10 Example

This example finds the eigenvalues in the half-open interval $(-1.0, 1.0]$, and corresponding eigenvectors, of the generalized symmetric eigenproblem $Az = \lambda Bz$, where

$$A = \begin{pmatrix} 0.24 & 0.39 & 0.42 & -0.16 \\ 0.39 & -0.11 & 0.79 & 0.63 \\ 0.42 & 0.79 & -0.25 & 0.48 \\ -0.16 & 0.63 & 0.48 & -0.03 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 4.16 & -3.12 & 0.56 & -0.10 \\ -3.12 & 5.03 & -0.83 & 1.09 \\ 0.56 & -0.83 & 0.76 & 0.34 \\ -0.10 & 1.09 & 0.34 & 1.18 \end{pmatrix}.$$

The example program for F08TCF (DSPGVD) illustrates solving a generalized symmetric eigenproblem of the form $ABz = \lambda z$.

10.1 Program Text

```
    Program f08tbfe

!     F08TBF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: dspgvx, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Real (Kind=nag_wp), Parameter    :: zero = 0.0E+0_nag_wp
      Integer, Parameter               :: nin = 5, nout = 6
      Character (1), Parameter         :: uplo = 'U'
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: abstol, vl, vu
      Integer                          :: i, ifail, il, info, iu, j, ldz, m, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: ap(:), bp(:), w(:), work(:), z(:,:)
      Integer, Allocatable             :: iwork(:), jfail(:)
!     .. Executable Statements ..
      Write (nout,*) 'F08TBF Example Program Results'
      Write (nout,*)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      ldz = n
      m = n
      Allocate (ap((n*(n+1))/2),bp((n*(n+1))/2),w(n),work(8*n),z(ldz,m),iwork( &
        5*n),jfail(n))

!     Read the lower and upper bounds of the interval to be searched,
!     and read the upper or lower triangular parts of the matrices A
!     and B from data file

      Read (nin,*) vl, vu
      If (uplo=='U') Then
        Read (nin,*)((ap(i+(j*(j-1))/2),j=i,n),i=1,n)
        Read (nin,*)((bp(i+(j*(j-1))/2),j=i,n),i=1,n)
      Else If (uplo=='L') Then
        Read (nin,*)((ap(i+((2*n-j)*(j-1))/2),j=1,i),i=1,n)
        Read (nin,*)((bp(i+((2*n-j)*(j-1))/2),j=1,i),i=1,n)
      End If

!     Set the absolute error tolerance for eigenvalues. With abstol
!     set to zero, the default value is used instead

      abstol = zero

!     Solve the generalized symmetric eigenvalue problem
!     A*x = lambda*B*x (itype = 1)

!     The NAG name equivalent of dspgvx is f08tbf
      Call dspgvx(1,'Vectors','Values in range',uplo,n,ap,bp,vl,vu,il,iu, &
        abstol,m,w,z,ldz,work,iwork,jfail,info)

      If (info>=0 .And. info<=n) Then

!       Print solution

        Write (nout,99999) 'Number of eigenvalues found =', m
        Write (nout,*)
        Write (nout,*) 'Eigenvalues'
        Write (nout,99998) w(1:m)
        Flush (nout)

!       Normalize the eigenvectors
        Do i = 1, m
          z(1:n,i) = z(1:n,i)/z(1,i)
```

```
        End Do

!       ifail: behaviour on error exit
!              =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
        ifail = 0
        Call x04caf('General',' ',n,m,z,ldz,'Selected eigenvectors',ifail)

        If (info>0) Then
          Write (nout,99999) 'INFO eigenvectors failed to converge, INFO =', &
            info
          Write (nout,*) 'Indices of eigenvectors that did not converge'
          Write (nout,99997) jfail(1:m)
        End If
      Else If (info>n .And. info<=2*n) Then
        i = info - n
        Write (nout,99996) 'The leading minor of order ', i, &
          ' of B is not positive definite'
      Else
        Write (nout,99999) 'Failure in DSPGVX. INFO =', info
      End If

99999 Format (1X,A,I5)
99998 Format (3X,(8F8.4))
99997 Format (3X,(8I8))
99996 Format (1X,A,I4,A)
    End Program f08tbfe
```

10.2 Program Data

```
F08TBF Example Program Data

  4                         :Value of N

 -1.0    1.0                :Values of VL and VU

  0.24   0.39   0.42  -0.16
        -0.11   0.79   0.63
               -0.25   0.48
                      -0.03  :End of matrix A

  4.16  -3.12   0.56  -0.10
         5.03  -0.83   1.09
                0.76   0.34
                       1.18  :End of matrix B
```

10.3 Program Results

```
 F08TBF Example Program Results

 Number of eigenvalues found =    2

 Eigenvalues
    -0.4548  0.1001
 Selected eigenvectors
            1          2
 1      1.0000     1.0000
 2      1.7303     0.0830
 3     -1.1354    -0.1129
 4     -2.0169    -1.0611
```