NAG Library Routine Document

F08TCF (DSPGVD)

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

F08TCF (DSPGVD) computes all the eigenvalues and, optionally, the 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 format, and B is also positive definite. If eigenvectors are desired, it uses a divide-and-conquer algorithm.

2 Specification

```
SUBROUTINE F08TCF (ITYPE, JOBZ, UPLO, N, AP, BP, W, Z, LDZ, WORK, LWORK,    &
                   IWORK, LIWORK, INFO)
INTEGER            ITYPE, N, LDZ, LWORK, IWORK(max(1,LIWORK)), LIWORK,      &
                   INFO
REAL (KIND=nag_wp) AP(*), BP(*), W(N), Z(LDZ,*), WORK(max(1,LWORK))
CHARACTER(1)       JOBZ, UPLO
```

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

3 Description

F08TCF (DSPGVD) 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 eigenvalues and, optionally, the 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

Golub G H and Van Loan C F (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

5 Arguments

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: 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'.

4: N – INTEGER *Input*

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

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

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

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

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 $\mathrm{AP}(i + j(j-1)/2)$ for $i \leq j$;

if UPLO = 'L', the lower triangle of A must be stored with element A_{ij} in $\mathrm{AP}(i + (2n-j)(j-1)/2)$ for $i \geq j$.

On exit: the contents of AP are destroyed.

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

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

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 $\mathrm{BP}(i + j(j-1)/2)$ for $i \leq j$;

if UPLO = 'L', the lower triangle of B must be stored with element B_{ij} in $\mathrm{BP}(i + (2n-j)(j-1)/2)$ for $i \geq j$.

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

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

On exit: the eigenvalues in ascending order.

8: Z(LDZ, $*$) – REAL (KIND=nag_wp) array *Output*

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

On exit: if JOBZ = 'V', Z contains the matrix Z of eigenvectors. 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 JOBZ = 'N', Z is not referenced.

9: LDZ – INTEGER *Input*

On entry: the first dimension of the array Z as declared in the (sub)program from which F08TCF (DSPGVD) is called.

Constraints:

if JOBZ = 'V', LDZ $\geq \max(1, \mathrm{N})$;
otherwise LDZ ≥ 1.

10: WORK($\max(1, \mathrm{LWORK})$) – REAL (KIND=nag_wp) array *Workspace*

On exit: if INFO $= 0$, WORK(1) contains the minimum value of LWORK required for optimal performance.

11: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08TCF (DSPGVD) is called.

If LWORK $= -1$, a workspace query is assumed; the routine only calculates the minimum sizes of the WORK and IWORK arrays, returns these values as the first entries of the WORK and IWORK arrays, and no error message related to LWORK or LIWORK is issued.

Constraints:

if $\mathrm{N} \leq 1$, LWORK ≥ 1;
if JOBZ = 'N' and $\mathrm{N} > 1$, LWORK $\geq 2 \times \mathrm{N}$;
if JOBZ = 'V' and $\mathrm{N} > 1$, LWORK $\geq 1 + 6 \times \mathrm{N} + 2 \times \mathrm{N}^2$.

12: IWORK($\max(1, \mathrm{LIWORK})$) – INTEGER array *Workspace*

On exit: if INFO $= 0$, IWORK(1) returns the minimum LIWORK.

13: LIWORK – INTEGER *Input*

On entry: the dimension of the array IWORK as declared in the (sub)program from which F08TCF (DSPGVD) is called.

If LIWORK $= -1$, a workspace query is assumed; the routine only calculates the minimum sizes of the WORK and IWORK arrays, returns these values as the first entries of the WORK and IWORK arrays, and no error message related to LWORK or LIWORK is issued.

Constraints:

if JOBZ = 'N' or N ≤ 1, LIWORK ≥ 1;
if JOBZ = 'V' and N > 1, LIWORK $\geq 3 + 5 \times$ N.

14: 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 > 0

F07GDF (DPPTRF) or F08GCF (DSPEVD) returned an error code:

$\leq$ N if INFO $= i$, F08GCF (DSPEVD) failed to converge; i off-diagonal elements of an intermediate tridiagonal form did not converge to zero;

$>$ N 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.

The example program below illustrates the computation of approximate error bounds.

8 Parallelism and Performance

F08TCF (DSPGVD) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F08TCF (DSPGVD) 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 F08TQF (ZHPGVD).

10 Example

This example finds all the eigenvalues and eigenvectors of the generalized symmetric eigenproblem $ABz = \lambda z$, where

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

together with an estimate of the condition number of B, and approximate error bounds for the computed eigenvalues and eigenvectors.

The example program for F08TAF (DSPGV) illustrates solving a generalized symmetric eigenproblem of the form $Az = \lambda Bz$.

10.1 Program Text

```
    Program f08tcfe

!     F08TCF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: dspgvd, dtpcon, f06rdf, nag_wp, x02ajf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
      Character (1), Parameter         :: uplo = 'U'
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: anorm, bnorm, eps, rcond, rcondb, t1
      Integer                          :: aplen, i, info, j, liwork, lwork, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: ap(:), bp(:), eerbnd(:), w(:),       &
                                          work(:)
      Real (Kind=nag_wp)               :: dummy(1,1)
      Integer                          :: idum(1)
      Integer, Allocatable             :: iwork(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: abs, max, nint
!     .. Executable Statements ..
      Write (nout,*) 'F08TCF Example Program Results'
      Write (nout,*)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      aplen = (n*(n+1))/2
      Allocate (ap(aplen),bp(aplen),eerbnd(n),w(n))

!     Use routine workspace query to get optimal workspace.
      lwork = -1
      liwork = -1
!     The NAG name equivalent of dspgvd is f08tcf
      Call dspgvd(2,'No vectors',uplo,n,ap,bp,w,dummy,n,dummy,lwork,idum,      &
        liwork,info)

!     Make sure that there is at least minimum workspace.
      lwork = max(3*n,nint(dummy(1,1)))
      liwork = max(n,idum(1))
      Allocate (work(lwork),iwork(liwork))

!     Read the upper or lower triangular parts of the matrices A and
!     B from data file

      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

!     Compute the one-norms of the symmetric matrices A and B

      anorm = f06rdf('One norm',uplo,n,ap,work)
      bnorm = f06rdf('One norm',uplo,n,bp,work)

!     Solve the generalized symmetric eigenvalue problem
!     A*B*x = lambda*x (itype = 2)

!     In the following call the 9th argument is set to n rather
!     than 1 to avoid an incorrect error message in some vendor
!     versions of LAPACK.
!     The NAG name equivalent of dspgvd is f08tcf
      Call dspgvd(2,'No vectors',uplo,n,ap,bp,w,dummy,n,work,lwork,iwork,      &
        liwork,info)

      If (info==0) Then

!       Print solution

        Write (nout,*) 'Eigenvalues'
        Write (nout,99999) w(1:n)

!       Call DTPCON (F07UGF) to estimate the reciprocal condition
!       number of the Cholesky factor of B.  Note that:
!       cond(B) = 1/rcond**2.  DTPCON requires WORK and IWORK to be
!       of length at least 3*n and n respectively

        Call dtpcon('One norm',uplo,'Non-unit',n,bp,rcond,work,iwork,info)

!       Print the reciprocal condition number of B

        rcondb = rcond**2
        Write (nout,*)
        Write (nout,*) 'Estimate of reciprocal condition number for B'
        Write (nout,99998) rcondb

!       Get the machine precision, eps, and if rcondb is not less
!       than eps**2, compute error estimates for the eigenvalues

        eps = x02ajf()
        If (rcond>=eps) Then
          t1 = anorm*bnorm
          Do i = 1, n
            eerbnd(i) = eps*(t1+abs(w(i))/rcondb)
          End Do

!         Print the approximate error bounds for the eigenvalues

          Write (nout,*)
          Write (nout,*) 'Error estimates for the eigenvalues'
          Write (nout,99998) eerbnd(1:n)
        Else
          Write (nout,*)
          Write (nout,*) 'B is very ill-conditioned, error ',                &
            'estimates have not been computed'
        End If
      Else If (info>n .And. info<=2*n) Then
        i = info - n
        Write (nout,99997) 'The leading minor of order ', i,                  &
          ' of B is not positive definite'
      Else
        Write (nout,99996) 'Failure in DSPGVD. INFO =', info
      End If
```

```
99999 Format (3X,(6F11.4))
99998 Format (4X,1P,6E11.1)
99997 Format (1X,A,I4,A)
99996 Format (1X,A,I4)
    End Program f08tcfe
```

10.2 Program Data

```
F08TCF Example Program Data

  4                         :Value of N

  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

```
 F08TCF Example Program Results

 Eigenvalues
       -3.5411    -0.3347     0.2983     2.2544

 Estimate of reciprocal condition number for B
        5.8E-03

 Error estimates for the eigenvalues
        7.0E-14    8.6E-15    7.9E-15    4.6E-14
```