NAG Fortran Library Routine Document

F08VEF (DGGSVP)

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

F08VEF (DGGSVP) uses orthogonal transformations to simultaneously reduce the m by n matrix A and the p by n matrix B to upper triangular form. This factorization is usually used as a preprocessing step for computing the generalized singular value decomposition (GSVD).

2 Specification

```
 SUBROUTINE F08VEF (JOBU, JOBV, JOBQ, M, P, N, A, LDA, B, LDB, TOLA,
1                   TOLB, K, L, U, LDU, V, LDV, Q, LDQ, IWORK, TAU, WORK,
2                   INFO)
 INTEGER            M, P, N, LDA, LDB, K, L, LDU, LDV, LDQ, IWORK(*),
1                   INFO
 double precision   A(LDA,*), B(LDB,*), TOLA, TOLB, U(LDU,*), V(LDV,*),
1                   Q(LDQ,*), TAU(*), WORK(*)
 CHARACTER*1        JOBU, JOBV, JOBQ
```

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

3 Description

F08VEF (DGGSVP) computes orthogonal matrices U, V and Q such that

$$U^{\mathrm{T}}AQ = \begin{cases} \begin{array}{r} \\ k \\ l \\ m-k-l \end{array} \begin{array}{c} \begin{array}{ccc} n-k-l & k & l \end{array} \\ \begin{pmatrix} 0 & A_{12} & A_{13} \\ 0 & 0 & A_{23} \\ 0 & 0 & 0 \end{pmatrix} \end{array} & \text{if } m-k-l \geq 0; \\ \begin{array}{r} \\ k \\ m-k \end{array} \begin{array}{c} \begin{array}{ccc} n-k-l & k & l \end{array} \\ \begin{pmatrix} 0 & A_{12} & A_{13} \\ 0 & 0 & A_{23} \end{pmatrix} \end{array} & \text{if } m-k-l < 0; \end{cases}$$

$$V^{\mathrm{T}}BQ = \begin{array}{r} \\ l \\ p-l \end{array} \begin{array}{c} \begin{array}{ccc} n-k-l & k & l \end{array} \\ \begin{pmatrix} 0 & 0 & B_{13} \\ 0 & 0 & 0 \end{pmatrix} \end{array}$$

where the k by k matrix A_{12} and l by l matrix B_{13} are non-singular upper triangular; A_{23} is l by l upper triangular if $m-k-l \geq 0$ and is $(m-k)$ by l upper trapezoidal otherwise. $(k+l)$ is the effective numerical rank of the $(m+p)$ by n matrix $\begin{pmatrix} A^{\mathrm{T}} & B^{\mathrm{T}} \end{pmatrix}^{\mathrm{T}}$.

This decomposition is usually used as the preprocessing step for computing the Generalized Singular Value Decomposition (GSVD), see routine F08VAF (DGGSVD).

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 URL: 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 Parameters

1: JOBU – CHARACTER*1 *Input*

On entry: if JOBU = 'U', the orthogonal matrix U is computed.

If JOBU = 'N', U is not computed.

2: JOBV – CHARACTER*1 *Input*

On entry: if JOBV = 'V', the orthogonal matrix V is computed.

If JOBV = 'N', V is not computed.

3: JOBQ – CHARACTER*1 *Input*

On entry: if JOBQ = 'Q', the orthogonal matrix Q is computed.

If JOBQ = 'N', Q is not computed.

4: M – INTEGER *Input*

On entry: m, the number of rows of the matrix A.

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

5: P – INTEGER *Input*

On entry: p, the number of rows of the matrix B.

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

6: N – INTEGER *Input*

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

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

7: A(LDA,$*$) – ***double precision*** array *Input/Output*

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

On entry: the m by n matrix A.

On exit: contains the triangular (or trapezoidal) matrix described in Section 3.

8: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08VEF (DGGSVP) is called.

Constraint: $\mathrm{LDA} \geq \max(1, \mathrm{M})$.

9: B(LDB,$*$) – ***double precision*** array *Input/Output*

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

On entry: the p by n matrix B.

On exit: contains the triangular matrix described in Section 3.

10: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F08VEF (DGGSVP) is called.

Constraint: $\mathrm{LDB} \geq \max(1, \mathrm{P})$.

11: TOLA – ***double precision*** *Input*
12: TOLB – ***double precision*** *Input*

On entry: TOLA and TOLB are the thresholds to determine the effective numerical rank of matrix B and a subblock of A. Generally, they are set to

$$\text{TOLA} = \max(\text{M}, \text{N})\|A\|\epsilon,$$
$$\text{TOLB} = \max(\text{P}, \text{N})\|B\|\epsilon,$$

where ϵ is the ***machine precision***.

The size of TOLA and TOLB may affect the size of backward errors of the decomposition.

13: K – INTEGER *Output*
14: L – INTEGER *Output*

On exit: K and L specify the dimension of the subblocks k and l as described in Section 3; $(k + l)$ is the effective numerical rank of $\left(\text{A}^{\text{T}} \quad \text{B}^{\text{T}}\right)^{\text{T}}$.

15: U(LDU,$*$) – ***double precision*** array *Output*

Note: the second dimension of the array U must be at least $\max(1, \text{M})$.

On exit: if JOBU = 'U', U contains the orthogonal matrix U.

If JOBU = 'N', U is not referenced.

16: LDU – INTEGER *Input*

On entry: the first dimension of the array U as declared in the (sub)program from which F08VEF (DGGSVP) is called.

Constraints:

if JOBU = 'U', LDU $\geq \max(1, \text{M})$;
LDU ≥ 1 otherwise.

17: V(LDV,$*$) – ***double precision*** array *Output*

Note: the second dimension of the array V must be at least $\max(1, \text{M})$.

On exit: if JOBV = 'V', V contains the orthogonal matrix V.

If JOBV = 'N', V is not referenced.

18: LDV – INTEGER *Input*

On entry: the first dimension of the array V as declared in the (sub)program from which F08VEF (DGGSVP) is called.

Constraints:

if JOBV = 'V', LDV $\geq \max(1, \text{P})$;
LDV ≥ 1 otherwise.

19: Q(LDQ,$*$) – ***double precision*** array *Output*

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

On exit: if JOBQ = 'Q', Q contains the orthogonal matrix Q.

If JOBQ = 'N', Q is not referenced.

20: LDQ – INTEGER *Input*

On entry: the first dimension of the array Q as declared in the (sub)program from which F08VEF (DGGSVP) is called.

Constraints:

if JOBQ = 'Q', LDQ $\geq \max(1, \mathrm{N})$;
LDQ ≥ 1 otherwise.

21: IWORK($*$) – INTEGER array *Workspace*

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

22: TAU($*$) – ***double precision*** array *Workspace*

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

23: WORK($*$) – ***double precision*** array *Workspace*

Note: the dimension of the array WORK must be at least $\max(1, 3 \times \mathrm{N}, \mathrm{M}, \mathrm{P})$.

24: INFO – INTEGER *Output*

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

6 Error Indicators and Warnings

Errors or warnings detected by the routine:

INFO < 0

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

7 Accuracy

The computed factorization is nearly the exact factorization for nearby matrices $(A + E)$ and $(B + F)$, where

$$\|E\|_2 = O(\epsilon)\|A\|_2 \quad \text{and} \quad \|F\|_2 = O(\epsilon)\|B\|_2,$$

and ϵ is the ***machine precision***.

8 Further Comments

The complex analogue of this routine is F08VSF (ZGGSVP).

9 Example

This example finds the generalized factorization

$$A = U\Sigma_1 \begin{pmatrix} 0 & S \end{pmatrix} Q^{\mathrm{T}}, \quad B = V\Sigma_2 \begin{pmatrix} 0 & T \end{pmatrix} Q^{\mathrm{T}},$$

of the matrix pair $\begin{pmatrix} A & B \end{pmatrix}$, where

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \\ 4 & 5 & 6 \\ 7 & 8 & 8 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} -2 & -3 & 3 \\ 4 & 6 & 5 \end{pmatrix}.$$

9.1 Program Text

```
*     F08VEF Example Program Text
*     Mark 21 Release. NAG Copyright 2004.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          MMAX, NMAX, PMAX
      PARAMETER        (MMAX=10,NMAX=10,PMAX=10)
```

```
      INTEGER          LDA, LDB, LDQ, LDU, LDV
      PARAMETER        (LDA=MMAX,LDB=PMAX,LDQ=NMAX,LDU=MMAX,LDV=PMAX)
*     .. Local Scalars ..
      DOUBLE PRECISION EPS, TOLA, TOLB
      INTEGER          I, IFAIL, INFO, IRANK, J, K, L, M, N, P
*     .. Local Arrays ..
      DOUBLE PRECISION A(LDA,NMAX), B(LDB,NMAX), Q(LDQ,NMAX), TAU(NMAX),
     +                 U(LDU,MMAX), V(LDV,PMAX), WORK(MMAX+3*NMAX+PMAX)
      INTEGER          IWORK(NMAX)
      CHARACTER        CLABS(1), RLABS(1)
*     .. External Functions ..
      DOUBLE PRECISION F06RAF, X02AJF
      EXTERNAL         F06RAF, X02AJF
*     .. External Subroutines ..
      EXTERNAL         DGGSVP, X04CBF
*     .. Intrinsic Functions ..
      INTRINSIC        MAX
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08VEF Example Program Results'
      WRITE (NOUT,*)
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) M, N, P
      IF (M.LE.MMAX .AND. N.LE.NMAX .AND. P.LE.PMAX) THEN
*
*        Read the m by n matrix A and p by n matrix B from data file
*
         READ (NIN,*) ((A(I,J),J=1,N),I=1,M)
         READ (NIN,*) ((B(I,J),J=1,N),I=1,P)
*
*        Compute TOLA and TOLB as
*           TOLA = max(M,N)*norm(A)*macheps
*           TOLB = max(P,N)*norm(B)*macheps
*
         EPS = X02AJF()
         TOLA = MAX(M,N)*F06RAF('One-norm',M,N,A,LDA,WORK)*EPS
         TOLB = MAX(P,N)*F06RAF('One-norm',P,N,B,LDB,WORK)*EPS
*
*        Compute the factorization of (A, B)
*           (A = U*S*(Q**T), B = V*T*(Q**T))
*
         CALL DGGSVP('U','V','Q',M,P,N,A,LDA,B,LDB,TOLA,TOLB,K,L,U,LDU,
     +               V,LDV,Q,LDQ,IWORK,TAU,WORK,INFO)
*
*        Print solution
*
         IRANK = K + L
         WRITE (NOUT,*) 'Numerical rank of (A**T B**T)**T (K+L)'
         WRITE (NOUT,99999) IRANK
*
         WRITE (NOUT,*)
         IF (M.GE.IRANK) THEN
            IFAIL = 0
            CALL X04CBF('Upper','Non-unit',IRANK,IRANK,A(1,N-IRANK+1),
     +                  LDA,'1P,E12.4','Upper triangular matrix S',
     +                  'Integer',RLABS,'Integer',CLABS,80,0,IFAIL)
         ELSE
            IFAIL = 0
            CALL X04CBF('Upper','Non-unit',M,IRANK,A(1,N-IRANK+1),LDA,
     +                  '1P,E12.4','Upper trapezoidal matrix S',
     +                  'Integer',RLABS,'Integer',CLABS,80,0,IFAIL)
         END IF
         WRITE (NOUT,*)
         IFAIL = 0
         CALL X04CBF('Upper','Non-unit',L,L,B(1,N-L+1),LDB,'1P,E12.4',
     +               'Upper triangular matrix T','Integer',RLABS,
     +               'Integer',CLABS,80,0,IFAIL)
         WRITE (NOUT,*)
         IFAIL = 0
         CALL X04CBF('General',' ',M,M,U,LDU,'1P,E12.4',
     +               'Orthogonal matrix U','Integer',RLABS,'Integer',
```

```
     +               CLABS,80,0,IFAIL)
         WRITE (NOUT,*)
         IFAIL = 0
         CALL X04CBF('General',' ',P,P,V,LDV,'1P,E12.4',
     +               'Orthogonal matrix V','Integer',RLABS,'Integer',
     +               CLABS,80,0,IFAIL)
         WRITE (NOUT,*)
         IFAIL = 0
         CALL X04CBF('General',' ',N,N,Q,LDQ,'1P,E12.4',
     +               'Orthogonal matrix Q','Integer',RLABS,'Integer',
     +               CLABS,80,0,IFAIL)
       ELSE
          WRITE (NOUT,*) 'MMAX and/or NMAX too small'
       END IF
       STOP
*
99999 FORMAT (1X,I5)
      END
```

9.2 Program Data

```
F08VEF Example Program Data

  4    3    2   :Values of M, N and P

  1.0  2.0  3.0
  3.0  2.0  1.0
  4.0  5.0  6.0
  7.0  8.0  8.0 :End of matrix A

 -2.0 -3.0  3.0
  4.0  6.0  5.0 :End of matrix B
```

9.3 Program Results

```
 F08VEF Example Program Results

 Numerical rank of (A**T B**T)**T (K+L)
     3

 Upper triangular matrix S
              1           2           3
 1  -2.0569E+00  1.0771E+01 -7.2814E+00
 2               7.1947E+00 -7.5262E+00
 3                          5.8129E-01

 Upper triangular matrix T
              1           2
 1   8.0623E+00 -3.1305E+00
 2              -4.9193E+00

 Orthogonal matrix U
              1           2           3           4
 1  -1.3484E-01  5.1025E-01 -2.4351E-01  8.1373E-01
 2   6.7420E-01 -5.4670E-01 -3.5349E-01  3.4874E-01
 3   2.6968E-01  4.8292E-01 -6.9127E-01 -4.6499E-01
 4   6.7420E-01  4.5558E-01  5.8129E-01  1.5127E-15

 Orthogonal matrix V
              1           2
 1  -4.4721E-01  8.9443E-01
 2   8.9443E-01  4.4721E-01

 Orthogonal matrix Q
              1           2           3
 1  -8.3205E-01  5.5470E-01  0.0000E+00
 2   5.5470E-01  8.3205E-01  0.0000E+00
 3   0.0000E+00  0.0000E+00 -1.0000E+00
```