NAG 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, TOLB,       &
                   K, L, U, LDU, V, LDV, Q, LDQ, IWORK, TAU, WORK, INFO)

INTEGER            M, P, N, LDA, LDB, K, L, LDU, LDV, LDQ, IWORK(N), INFO
REAL (KIND=nag_wp) A(LDA,*), B(LDB,*), TOLA, TOLB, U(LDU,*), V(LDV,*),          &
                   Q(LDQ,*), TAU(N), WORK(max(3*N,M,P))
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

$$U^{\mathrm{T}}AQ = \begin{cases} \begin{array}{c c} & \begin{array}{ccc} n-k-l & k & l \end{array} \\ \begin{array}{r} k \\ l \\ m-k-l \end{array} & \begin{pmatrix} 0 & A_{12} & A_{13} \\ 0 & 0 & A_{23} \\ 0 & 0 & 0 \end{pmatrix} \end{array}, \text{ if } m-k-l \geq 0; \\[2em] \begin{array}{c c} & \begin{array}{ccc} n-k-l & k & l \end{array} \\ \begin{array}{r} k \\ m-k \end{array} & \begin{pmatrix} 0 & A_{12} & A_{13} \\ 0 & 0 & A_{23} \end{pmatrix} \end{array}, \text{ if } m-k-l < 0; \end{cases}$$

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

where the k by k matrix A_{12} and l by l matrix B_{13} are nonsingular 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 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.

Constraint: JOBU = 'U' or 'N'.

2: JOBV – CHARACTER(1) *Input*

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

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

Constraint: JOBV = 'V' or 'N'.

3: JOBQ – CHARACTER(1) *Input*

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

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

Constraint: JOBQ = 'Q' or 'N'.

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,$*$) – REAL (KIND=nag_wp) 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,$*$) – REAL (KIND=nag_wp) 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: $\text{LDB} \geq \max(1, \text{P})$.

11: TOLA – REAL (KIND=nag_wp) *Input*
12: TOLB – REAL (KIND=nag_wp) *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,$*$) – REAL (KIND=nag_wp) array *Output*

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

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', $\text{LDU} \geq \max(1, \text{M})$;
otherwise $\text{LDU} \geq 1$.

17: V(LDV,$*$) – REAL (KIND=nag_wp) array *Output*

Note: the second dimension of the array V must be at least $\max(1, \text{P})$ if JOBV = 'V', and at least 1 otherwise.

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', $\text{LDV} \geq \max(1, \text{P})$;
otherwise $\text{LDV} \geq 1$.

19: Q(LDQ,$*$) – REAL (KIND=nag_wp) array *Output*

Note: the second dimension of the array Q must be at least $\max(1, N)$ if JOBQ = 'Q', and at least 1 otherwise.

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 $\geq \max(1, N)$;
otherwise LDQ ≥ 1.

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

22: TAU(N) – REAL (KIND=nag_wp) array *Workspace*

23: WORK($\max(3 \times N, M, P)$) – REAL (KIND=nag_wp) array *Workspace*

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 $= -i$, argument i 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

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

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

```
    Program f08vefe

!     F08VEF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: dggsvp, f06raf, nag_wp, x02ajf, x04cbf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter              :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)              :: eps, tola, tolb
      Integer                         :: i, ifail, info, irank, k, l, lda,    &
                                         ldb, ldq, ldu, ldv, m, n, p
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable :: a(:,:), b(:,:), q(:,:), tau(:),      &
                                         u(:,:), v(:,:), work(:)
      Integer, Allocatable            :: iwork(:)
      Character (1)                   :: clabs(1), rlabs(1)
!     .. Intrinsic Procedures ..
      Intrinsic                       :: max, real
!     .. Executable Statements ..
      Write (nout,*) 'F08VEF Example Program Results'
      Write (nout,*)
      Flush (nout)

!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) m, n, p
      lda = m
      ldb = p
      ldq = n
      ldu = m
      ldv = p
      Allocate (a(lda,n),b(ldb,n),q(ldq,n),tau(n),u(ldu,m),v(ldv,p), &
        work(m+3*n+p),iwork(n))

!     Read the m by n matrix A and p by n matrix B from data file

      Read (nin,*)(a(i,1:n),i=1,m)
      Read (nin,*)(b(i,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 = real(max(m,n),kind=nag_wp)*f06raf('One-norm',m,n,a,lda,work)*eps
      tolb = real(max(p,n),kind=nag_wp)*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))

!     The NAG name equivalent of dggsvp is f08vef
      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,*)
      Flush (nout)
      If (m>=irank) Then
```

```
!       ifail: behaviour on error exit
!              =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
        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,*)
      Flush (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,*)
      Flush (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,*)
      Flush (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,*)
      Flush (nout)

      ifail = 0
      Call x04cbf('General',' ',n,n,q,ldq,'1P,E12.4','Orthogonal matrix Q', &
        'Integer',rlabs,'Integer',clabs,80,0,ifail)

99999 Format (1X,I5)
    End Program f08vefe
```

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
```