NAG Fortran Library Routine Document

F08VAF (DGGSVD)

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

F08VAF (DGGSVD) computes the generalized singular value decomposition (GSVD) of an m by n real matrix A and a p by n real matrix B.

2 Specification

```
      SUBROUTINE F08VAF (JOBU, JOBV, JOBQ, M, N, P, K, L, A, LDA, B, LDB,
     1                   ALPHA, BETA, U, LDU, V, LDV, Q, LDQ, WORK, IWORK,
     2                   INFO)
      INTEGER            M, N, P, K, L, LDA, LDB, LDU, LDV, LDQ, IWORK(*),
     1                   INFO
      double precision   A(LDA,*), B(LDB,*), ALPHA(*), BETA(*), U(LDU,*),
     1                   V(LDV,*), Q(LDQ,*), WORK(*)
      CHARACTER*1        JOBU, JOBV, JOBQ
```

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

3 Description

The generalized singular value decomposition is given by

$$U^TAQ = D_1(\,0 \quad R\,), \quad V^TBQ = D_2(\,0 \quad R\,),$$

where U, V and Q are orthogonal matrices. Let $k+l$ be the effective numerical rank of the matrix $\begin{pmatrix} A^T & B^T \end{pmatrix}^T$, then R is a $k+l$ by $k+l$ nonsingular upper triangular matrix, D_1 and D_2 are m by $(k+l)$ and p by $(k+l)$ 'diagonal' matrices structured as follows:

if $m-k-l \geq 0$,

$$D_1 = \begin{array}{r} \\ k \\ l-l \\ m-k-l \end{array} \begin{array}{c} \begin{array}{cc} k & l \end{array} \\ \begin{pmatrix} I & 0 \\ 0 & C \\ 0 & 0 \end{pmatrix} \end{array}$$

$$D_2 = \begin{array}{r} \\ l \\ p-l \end{array} \begin{array}{c} \begin{array}{cc} k & l \end{array} \\ \begin{pmatrix} 0 & S \\ 0 & 0 \end{pmatrix} \end{array}$$

$$(\,0 \quad R\,) = \begin{array}{r} \\ k \\ l \end{array} \begin{array}{c} \begin{array}{ccc} n-k-l & k & l \end{array} \\ \begin{pmatrix} 0 & R_{11} & R_{12} \\ 0 & 0 & R_{22} \end{pmatrix} \end{array}$$

where

$$C = \mathrm{diag}(\alpha_{k+1}, \ldots, \alpha_{k+l}),$$
$$S = \mathrm{diag}(\beta_{k+1}, \ldots, \beta_{k+l}),$$

and

$$C^2 + S^2 = I.$$

R is stored in $\mathrm{A}(1:k+1, n-k-l+1:\mathrm{n})$ on exit.

If $m - k - l < 0$,

$$D_1 = \begin{array}{c} \\ k \\ m-k \end{array} \begin{array}{c} \begin{array}{ccc} k & m-k & k+l-m \end{array} \\ \begin{pmatrix} I & 0 & 0 \\ 0 & C & 0 \end{pmatrix} \end{array}$$

$$D_2 = \begin{array}{c} \\ m-k \\ k+l-m \\ p-l \end{array} \begin{array}{c} \begin{array}{ccc} k & m-k & k+l-m \end{array} \\ \begin{pmatrix} 0 & S & 0 \\ 0 & 0 & I \\ 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$\begin{pmatrix} 0 & R \end{pmatrix} = \begin{array}{c} \\ k \\ m-k \\ k+l-m \end{array} \begin{array}{c} \begin{array}{cccc} n-k-l & k & m-k & k+l-m \end{array} \\ \begin{pmatrix} 0 & R_{11} & R_{12} & R_{13} \\ 0 & 0 & R_{22} & R_{23} \\ 0 & 0 & 0 & R_{33} \end{pmatrix} \end{array}$$

where

$$C = \mathrm{diag}(\alpha_{k+1}, \ldots, \alpha_m),$$
$$S = \mathrm{diag}(\beta_{k+1}, \ldots, \beta_m),$$

and

$$C^2 + S^2 = I.$$

$\begin{pmatrix} R_{11} & R_{12} & R_{13} \\ 0 & R_{22} & R_{23} \end{pmatrix}$ is stored in $\mathrm{A}(1:m, n-k-l+1:n)$, and R_{33} is stored in $\mathrm{B}(m-k+1:l, n+m-k-l+1:n)$ on exit.

The routine computes C, S, R, and optionally the orthogonal transformation matrices U, V and Q.

In particular, if B is an n by n nonsingular matrix, then the GSVD of A and B implicitly gives the SVD of AB^{-1}:

$$AB^{-1} = U\left(D_1 D_2^{-1}\right)V^T.$$

If $\begin{pmatrix} A^T & B^T \end{pmatrix}^T$ has orthonormal columns, then the GSVD of A and B is also equal to the CS decomposition of A and B. Furthermore, the GSVD can be used to derive the solution of the eigenvalue problem:

$$A^T Ax = \lambda B^T Bx.$$

In some literature, the GSVD of A and B is presented in the form

$$U^T AX = \begin{pmatrix} 0 & D_1 \end{pmatrix}, \quad V^T BX = \begin{pmatrix} 0 & D_2 \end{pmatrix},$$

where U and V are orthogonal and X is nonsingular, D_1 and D_2 are 'diagonal'. The former GSVD form can be converted to the latter form by taking the nonsingular matrix X as

$$X = Q\begin{pmatrix} I & 0 \\ 0 & R^{-1} \end{pmatrix}.$$

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: N – INTEGER *Input*

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

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

6: P – INTEGER *Input*

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

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

7: K – INTEGER *Output*
8: L – INTEGER *Output*

On exit: K and L specify the dimension of the subblocks k and l as described in Section 3.

9: 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 matrix R, or part of R. See Section 3 for details.

10: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08VAF (DGGSVD) is called.

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

11: 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 R if $m - k - l < 0$. See Section 3 for details.

12: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F08VAF (DGGSVD) is called.

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

13: ALPHA($*$) – ***double precision*** array *Output*

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

On exit: see the description of BETA below.

14: BETA($*$) – ***double precision*** array *Output*

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

On exit: ALPHA and BETA contain the generalized singular value pairs of A and B, α_i and β_i;

$\mathrm{ALPHA}(1:\mathrm{K}) = 1$,

$\mathrm{BETA}(1:\mathrm{K}) = 0$,

and if $m - k - l \geq 0$,

$\mathrm{ALPHA}(\mathrm{K}+1:\mathrm{K}+\mathrm{L}) = C$,

$\mathrm{BETA}(\mathrm{K}+1:\mathrm{K}+\mathrm{L}) = S$,

or if $m - k - l < 0$,

$\mathrm{ALPHA}(\mathrm{K}+1:\mathrm{M}) = C$,

$\mathrm{ALPHA}(\mathrm{M}+1:\mathrm{K}+\mathrm{L}) = 0$,

$\mathrm{BETA}(\mathrm{K}+1:\mathrm{M}) = S$,

$\mathrm{BETA}(\mathrm{M}+1:\mathrm{K}+\mathrm{L}) = 1$, and

$\mathrm{ALPHA}(\mathrm{K}+\mathrm{L}+1:\mathrm{N}) = 0$,

$\mathrm{BETA}(\mathrm{K}+\mathrm{L}+1:\mathrm{N}) = 0$.

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

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

On exit: if JOBU $=$ 'U', U contains the m by m 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 F08VAF (DGGSVD) is called.

Constraints:

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

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

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

On exit: if JOBV $=$ 'V', V contains the p by p 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 F08VAF (DGGSVD) is called.

Constraints:

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

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

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

On exit: if JOBQ = 'Q', Q contains the n by n 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 F08VAF (DGGSVD) is called.

Constraints:

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

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

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

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

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

On exit: stores the sorting information. More precisely, the following loop will sort ALPHA

```
for I=K+1, min(M,K+L)  swap ALPHA(I) and ALPHA(IWORK(I))  endfor
```

such that $\text{ALPHA}(1) \geq \text{ALPHA}(2) \geq \cdots \geq \text{ALPHA}(N)$.

23: 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 argument had an illegal value.

INFO > 0

If INFO = 1, the Jacobi-type procedure failed to converge.

7 Accuracy

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

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

and ϵ is the ***machine precision***. See Section 4.12 of Anderson *et al.* (1999) for further details.

8 Further Comments

The complex analogue of this routine is F08VNF (ZGGSVD).

9 Example

To find the generalized singular value decomposition

$$A = U\Sigma_1(0 \quad R)Q^T, \quad B = V\Sigma_2(0 \quad R)Q^T,$$

where

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \\ 4 & 5 & 6 \\ 7 & 8 & 8 \end{pmatrix}$$

and

$$B = \begin{pmatrix} -2 & -3 & 3 \\ 4 & 6 & 5 \end{pmatrix},$$

together with estimates for the condition number of R and the error bound for the computed generalized singular values.

The example program assumes that $m \geq n$, and would need slight modification if this is not the case.

9.1 Program Text

Note: the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     F08VAF 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, RCOND, SERRBD
      INTEGER          I, IFAIL, INFO, IRANK, J, K, L, M, N, P
*     .. Local Arrays ..
      DOUBLE PRECISION A(LDA,NMAX), ALPHA(NMAX), B(LDB,NMAX),
     +                 BETA(NMAX), Q(LDQ,NMAX), U(LDU,MMAX),
     +                 V(LDV,PMAX), WORK(MMAX+3*NMAX)
      INTEGER          IWORK(NMAX)
      CHARACTER        CLABS(1), RLABS(1)
*     .. External Functions ..
      DOUBLE PRECISION X02AJF
      EXTERNAL         X02AJF
*     .. External Subroutines ..
      EXTERNAL         DGGSVD, DTRCON, X04CBF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08VAF 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 the generalized singular value decomposition of (A, B)
*        (A = U*D1*(0 R)*(Q**T), B = V*D2*(0 R)*(Q**T), m.ge.n)
*
         CALL DGGSVD('U','V','Q',M,N,P,K,L,A,LDA,B,LDB,ALPHA,BETA,U,LDU,
     +               V,LDV,Q,LDQ,WORK,IWORK,INFO)
```

```
*
         IF (INFO.EQ.0) THEN
*
*           Print solution
*
            IRANK = K + L
            WRITE (NOUT,*)
     +        'Number of infinite generalized singular values (K)'
            WRITE (NOUT,99999) K
            WRITE (NOUT,*)
     +        'Number of finite generalized singular values (L)'
            WRITE (NOUT,99999) L
            WRITE (NOUT,*) 'Numerical rank of (A**T B**T)**T (K+L)'
            WRITE (NOUT,99999) IRANK
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Finite generalized singular values'
            WRITE (NOUT,99998) (ALPHA(J)/BETA(J),J=K+1,IRANK)
*
            IFAIL = 0
            WRITE (NOUT,*)
            CALL X04CBF('General',' ',M,M,U,LDU,'1P,E12.4',
     +                  'Orthogonal matrix U','Integer',RLABS,'Integer',
     +                  CLABS,80,0,IFAIL)
            WRITE (NOUT,*)
            CALL X04CBF('General',' ',P,P,V,LDV,'1P,E12.4',
     +                  'Orthogonal matrix V','Integer',RLABS,'Integer',
     +                  CLABS,80,0,IFAIL)
            WRITE (NOUT,*)
            CALL X04CBF('General',' ',N,N,Q,LDQ,'1P,E12.4',
     +                  'Orthogonal matrix Q','Integer',RLABS,'Integer',
     +                  CLABS,80,0,IFAIL)
            WRITE (NOUT,*)
            CALL X04CBF('Upper triangular','Non-unit',IRANK,IRANK,
     +                  A(1,N-IRANK+1),LDA,'1P,E12.4',
     +                  'Non singular upper triangular matrix R',
     +                  'Integer',RLABS,'Integer',CLABS,80,0,IFAIL)
*
*           Call DTRCON (F07TGF) to estimate the reciprocal condition
*           number of R
*
            CALL DTRCON('Infinity-norm','Upper','Non-unit',IRANK,
     +                  A(1,N-IRANK+1),LDA,RCOND,WORK,IWORK,INFO)
*
            WRITE (NOUT,*)
            WRITE (NOUT,*)
     +        'Estimate of reciprocal condition number for R'
            WRITE (NOUT,99997) RCOND
            WRITE (NOUT,*)
*
*           So long as IRANK = N, get the machine precision, EPS, and
*           compute the approximate error bound for the computed
*           generalized singular values
*
            IF (IRANK.EQ.N) THEN
               EPS = X02AJF()
               SERRBD = EPS/RCOND
               WRITE (NOUT,*)
     +           'Error estimate for the generalized singular values'
               WRITE (NOUT,99997) SERRBD
            ELSE
               WRITE (NOUT,*) '(A**T B**T)**T is not of full rank'
            END IF
         ELSE
            WRITE (NOUT,99996) 'Failure in DGGSVD. INFO =', INFO
         END IF
      ELSE
         WRITE (NOUT,*) 'MMAX and/or NMAX too small'
      END IF
      STOP
*
99999 FORMAT (1X,I5)
```

```
99998 FORMAT (3X,8(1P,E12.4))
99997 FORMAT (1X,1P,E11.1)
99996 FORMAT (1X,A,I4)
      END
```

9.2 Program Data

```
F08VAF 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

```
 F08VAF Example Program Results

 Number of infinite generalized singular values (K)
     1
 Number of finite generalized singular values (L)
     2
 Numerical rank of (A**T B**T)**T (K+L)
     3

 Finite generalized singular values
     1.3151E+00  8.0185E-02

 Orthogonal matrix U
              1           2           3           4
 1  -1.3484E-01  5.2524E-01 -2.0924E-01  8.1373E-01
 2   6.7420E-01 -5.2213E-01 -3.8886E-01  3.4874E-01
 3   2.6968E-01  5.2757E-01 -6.5782E-01 -4.6499E-01
 4   6.7420E-01  4.1615E-01  6.1014E-01 -2.1473E-15

 Orthogonal matrix V
              1           2
 1   3.5539E-01 -9.3472E-01
 2   9.3472E-01  3.5539E-01

 Orthogonal matrix Q
              1           2           3
 1  -8.3205E-01 -9.4633E-02 -5.4657E-01
 2   5.5470E-01 -1.4195E-01 -8.1985E-01
 3   1.8475E-16 -9.8534E-01  1.7060E-01

 Non singular upper triangular matrix R
              1           2           3
 1  -2.0569E+00 -9.0121E+00 -9.3705E+00
 2              -1.0882E+01 -7.2688E+00
 3                          -6.0405E+00

 Estimate of reciprocal condition number for R
     4.2E-02

 Error estimate for the generalized singular values
     2.6E-15
```