NAG 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. F08VAF (DGGSVD) is marked as *deprecated* by LAPACK; the replacement routine is F08VCF (DGGSVD3) which makes better use of level 3 BLAS.

2 Specification

```
SUBROUTINE F08VAF (JOBU, JOBV, JOBQ, M, N, P, K, L, A, LDA, B, LDB,        &
                   ALPHA, BETA, U, LDU, V, LDV, Q, LDQ, WORK, IWORK,       &
                   INFO)

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

$$U^{\mathrm{T}}AQ = D_1\begin{pmatrix}0 & R\end{pmatrix}, \quad V^{\mathrm{T}}BQ = D_2\begin{pmatrix}0 & R\end{pmatrix},$$

where U, V and Q are orthogonal matrices. Let $(k+l)$ be the effective numerical rank of the matrix $\begin{pmatrix}A\\B\end{pmatrix}$, 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 $m-k-l \geq 0$,

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

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

$$\begin{pmatrix}0 & R\end{pmatrix} = \begin{array}{c} \\ k \\ l \end{array}\!\!\begin{array}{c} \begin{array}{ccc} n-k-l & k & l \end{array} \\ \begin{pmatrix} 0 & R_{11} & R_{12} \\ 0 & 0 & R_{22} \end{pmatrix} \end{array}$$

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 as a submatrix of A with elements R_{ij} stored as $A_{i,n-k-l+j}$ on exit.

If $m-k-l<0$,

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

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

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

where

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

and

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

$\begin{pmatrix} R_{11} & R_{12} & R_{13} \\ 0 & R_{22} & R_{23} \end{pmatrix}$ is stored as a submatrix of A with R_{ij} stored as $A_{i,n-k-l+j}$, and R_{33} is stored as a submatrix of B with $(R_{33})_{ij}$ stored as $B_{m-k+i,n+m-k-l+j}$.

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}:

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

If $\begin{pmatrix} A \\ B \end{pmatrix}$ 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:

$$A^{\mathrm{T}}Ax = \lambda B^{\mathrm{T}}Bx.$$

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

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

where U and V are orthogonal and X is nonsingular, and 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 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: 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: M ≥ 0.

5: N – INTEGER *Input*

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

Constraint: N ≥ 0.

6: P – INTEGER *Input*

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

Constraint: P ≥ 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; $(k+l)$ is the effective numerical rank of $\begin{pmatrix} A \\ B \end{pmatrix}$.

9: A(LDA, $*$) – REAL (KIND=nag_wp) array *Input/Output*

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

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

13: ALPHA(N) – REAL (KIND=nag_wp) array *Output*

On exit: see the description of BETA.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

The notation ALPHA(K : N) above refers to consecutive elements ALPHA(i), for $i = \mathrm{K}, \ldots, \mathrm{N}$.

15: U(LDU, $*$) – REAL (KIND=nag_wp) array *Output*

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

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 $\geq \max(1, \mathrm{M})$;
otherwise LDU ≥ 1.

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

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

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 $\geq \max(1, \mathrm{P})$;
otherwise LDV ≥ 1.

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

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

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

21: WORK($\max(3 \times \mathrm{N}, \mathrm{M}, \mathrm{P}) + \mathrm{N}$) – REAL (KIND=nag_wp) array *Workspace*

22: IWORK(N) – INTEGER array *Output*

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 ALPHA(1) $\geq$ ALPHA(2) $\geq \cdots \geq$ ALPHA(N).

23: 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 $= 1$

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 Parallelism and Performance

F08VAF (DGGSVD) 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 complex analogue of this routine is F08VNF (ZGGSVD).

10 Example

This example finds the generalized singular value decomposition

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

where

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

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.

10.1 Program Text

```
    Program f08vafe

!     F08VAF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: dggsvd, dtrcon, nag_wp, x02ajf, x04cbf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: eps, rcond, serrbd
      Integer                          :: i, ifail, info, irank, j, k, l, lda, &
                                          ldb, ldq, ldu, ldv, m, n, p
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:), alpha(:), b(:,:), beta(:),   &
```

```
                                       q(:,:), u(:,:), v(:,:), work(:)
      Integer, Allocatable             :: iwork(:)
      Character (1)                    :: clabs(1), rlabs(1)
!     .. Executable Statements ..
      Write (nout,*) 'F08VAF 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),alpha(n),b(ldb,n),beta(n),q(ldq,n),u(ldu,m),v(ldv,p), &
        work(m+3*n),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 the generalized singular value decomposition of (A, B)
!     (A = U*D1*(0 R)*(Q**T), B = V*D2*(0 R)*(Q**T), m>=n)
!     The NAG name equivalent of dggsvd is f08vaf
      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==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)

        Write (nout,*)
        Flush (nout)

!       ifail: behaviour on error exit
!              =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
        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)

        Call x04cbf('General',' ',p,p,v,ldv,'1P,E12.4','Orthogonal matrix V',  &
          'Integer',rlabs,'Integer',clabs,80,0,ifail)

        Write (nout,*)
        Flush (nout)

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

        Write (nout,*)
        Flush (nout)

        Call x04cbf('Upper triangular','Non-unit',irank,irank,a(1,n-irank+1),  &
          lda,'1P,E12.4','Nonsingular 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==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

99999 Format (1X,I5)
99998 Format (3X,8(1P,E12.4))
99997 Format (1X,1P,E11.1)
99996 Format (1X,A,I4)
    End Program f08vafe
```

10.2 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
```

10.3 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  1.5127E-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   0.0000E+00 -9.8534E-01  1.7060E-01

 Nonsingular 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
```