NAG Fortran Library Routine Document

F08YEF (DTGSJA)

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

F08YEF (DTGSJA) computes the generalized singular value decomposition (GSVD) of two real upper trapezoidal matrices A and B, where A is an m by n matrix and B is a p by n matrix.

A and B are assumed to be in the form returned by F08VEF (DGGSVP).

2 Specification

```
 SUBROUTINE F08YEF (JOBU, JOBV, JOBQ, M, P, N, K, L, A, LDA, B, LDB,
1                   TOLA, TOLB, ALPHA, BETA, U, LDU, V, LDV, Q, LDQ,
2                   WORK, NCYCLE, INFO)
 INTEGER            M, P, N, K, L, LDA, LDB, LDU, LDV, LDQ, NCYCLE, INFO
 double precision   A(LDA,*), B(LDB,*), TOLA, TOLB, ALPHA(*), BETA(*),
1                   U(LDU,*), V(LDV,*), Q(LDQ,*), WORK(*)
 CHARACTER*1        JOBU, JOBV, JOBQ
```

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

3 Description

F08YEF (DTGSJA) computes the GSVD of the matrices A and B which are assumed to have the form as returned by F08VEF (DGGSVP)

$$A = \begin{cases} \begin{array}{c} \\ k \\ l \\ m-k-l \end{array} \begin{array}{c} \begin{array}{ccc} n-k-l & k & l \end{array} \\ \begin{pmatrix} 0 & A_{12} & A_{13} \\ 0 & 0 & A_{23} \\ 0 & 0 & 0 \end{pmatrix} \end{array}, & \text{if } m-k-l \geq 0; \\ \begin{array}{c} \\ k \\ m-k \end{array} \begin{array}{c} \begin{array}{ccc} n-k-l & k & l \end{array} \\ \begin{pmatrix} 0 & A_{12} & A_{13} \\ 0 & 0 & A_{23} \end{pmatrix} \end{array}, & \text{if } m-k-l < 0; \end{cases}$$

$$B = \begin{array}{c} \\ l \\ p-l \end{array} \begin{array}{c} \begin{array}{ccc} n-k-l & k & l \end{array} \\ \begin{pmatrix} 0 & 0 & B_{13} \\ 0 & 0 & 0 \end{pmatrix} \end{array},$$

where the k by k matrix A_{12} and the 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.

F08YEF (DTGSJA) computes orthogonal matrices Q, U and V, diagonal matrices D_1 and D_2, and an upper triangular matrix R such that

$$U^{\mathrm{T}}AQ = D_1(0 \quad R), \quad V^{\mathrm{T}}BQ = D_2(0 \quad R).$$

Optionally Q, U and V may or may not be computed, or they may be premultiplied by matrices Q_1, U_1 and V_1 respectively.

If $(m - k - l) \geq 0$ then D_1, D_2 and R have the form

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

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

$$R = \begin{array}{r} \\ k \\ l \end{array} \begin{array}{c} \begin{array}{cc} k & l \end{array} \\ \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix} \end{array},$$

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

If $(m - k - l) < 0$ then D_1, D_2 and R have the form

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

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

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

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

In both cases the diagonal matrix C has non-negative diagonal elements, the diagonal matrix S has positive diagonal elements, so that S is non-singular, and $C^2 + S^2 = 1$. See Anderson *et al.* (1999) (Section 2.3.5.3) for further information.

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', U must contain an orthogonal matrix U_1 on entry, and the product U_1U is returned.

If JOBU = 'I', U is initialized to the unit matrix, and the orthogonal matrix U is returned.

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

2: JOBV – CHARACTER*1 *Input*

On entry: if JOBV = 'V', V must contain an orthogonal matrix V_1 on entry, and the product $V_1 V$ is returned.

If JOBV = 'I', V is initialized to the unit matrix, and the orthogonal matrix V is returned.

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

3: JOBQ – CHARACTER*1 *Input*

On entry: if JOBQ = 'Q', Q must contain an orthogonal matrix Q_1 on entry, and the product $Q_1 Q$ is returned.

If JOBQ = 'I', Q is initialized to the unit matrix, and the orthogonal matrix Q is returned.

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: K – INTEGER *Input*
8: L – INTEGER *Input*

On entry: K and L specify the sizes, k and l, of the subblocks of A and B, whose GSVD is to be computed by F08YEF (DTGSJA).

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: if $m - k - l \geq 0$, $\mathrm{A}(1 : k + l, n - k - l + 1 : n)$ contains the $(k + l)$ by $(k + l)$ upper triangular matrix R.

If $m - k - l < 0$, $\mathrm{A}(1 : m, n - k - l + 1 : n)$ contains the first m rows of the $(k + l)$ by $(k + l)$ upper triangular matrix R, and the submatrix R_{33} is returned in $\mathrm{B}(m - k + 1 : l, n + m - k - l + 1 : n)$.

10: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08YEF (DTGSJA) 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: if $m - k - l < 0$, $\mathrm{B}(m - k + 1 : l, n + m - k - l + 1 : n)$ contains the submatrix R_{33} of R.

12: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F08YEF (DTGSJA) is called.

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

13: TOLA – ***double precision*** *Input*
14: TOLB – ***double precision*** *Input*

On entry: TOLA and TOLB are the convergence criteria for the Jacobi-Kogbetliantz iteration procedure. Generally, they should be the same as used in the preprocessing step performed by F08VSF (ZGGSVP), say

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

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

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

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

On exit: see the description of BETA.

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

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

On exit: ALPHA and BETA contain the generalized singular value pairs of A and B:

ALPHA(i) = 1, BETA(i) = 0, for $i = 1, 2, \ldots, k$, and
if $m - k - l \geq 0$, ALPHA(i) = α_i, BETA(i) = β_i, for $i = k+1, k+2, \ldots, k+l$, or
if $m - k - l < 0$, ALPHA(i) = α_i, BETA(i) = β_i, for $i = k+1, k+2, \ldots, m$ and
ALPHA(i) = 0, BETA(i) = 1, for $i = m+1, m+2, \ldots, k+l$.

Furthermore, if $k + l < n$, ALPHA(i) = BETA(i) = 0, for $i = k+l+1, k+l+2, \ldots, n$.

17: U(LDU,$*$) – ***double precision*** array *Input/Output*

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

On entry: if JOBU = 'U', U must contain an m by m matrix U_1 (usually the orthogonal matrix returned by F08VEF (DGGSVP)).

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

If JOBU = 'U', U contains the product U_1U.

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

18: LDU – INTEGER *Input*

On entry: the first dimension of the array U as declared in the (sub)program from which F08YEF (DTGSJA) is called.

Constraints:

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

19: V(LDV,$*$) – ***double precision*** array *Input/Output*

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

On entry: if JOBV = 'V', V must contain an p by p matrix V_1 (usually the orthogonal matrix returned by F08VEF (DGGSVP)).

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

If JOBV = 'V', V contains the product $V_1 V$.

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

20: LDV – INTEGER *Input*

On entry: the first dimension of the array V as declared in the (sub)program from which F08YEF (DTGSJA) is called.

Constraints:

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

21: Q(LDQ,$*$) – ***double precision*** array *Input/Output*

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

On entry: if JOBQ = 'Q', Q must contain an n by n matrix Q_1 (usually the orthogonal matrix returned by F08VEF (DGGSVP)).

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

If JOBQ = 'Q', Q contains the product $Q_1 Q$.

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

22: LDQ – INTEGER *Input*

On entry: the first dimension of the array Q as declared in the (sub)program from which F08YEF (DTGSJA) is called.

Constraints:

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

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

Note: the dimension of the array WORK must be at least max(1, $2 \times$ N).

24: NCYCLE – INTEGER *Output*

On exit: the number of cycles required for convergence.

25: 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.

INFO = 1

The procedure does not converge after 40 cycles.

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

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

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

8 Further Comments

The complex analogue of this routine is F08YSF (ZTGSJA).

9 Example

This example finds the generalized singular value decomposition

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

of the matrix pair (A,B), 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

```
*     F08YEF 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, NCYCLE, P
*     .. Local Arrays ..
      DOUBLE PRECISION A(LDA,NMAX), ALPHA(NMAX), B(LDB,NMAX),
     +                 BETA(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, DTGSJA, X04CBF
*     .. Intrinsic Functions ..
      INTRINSIC        MAX
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08YEF 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 = U1*S*(Q1**T), B = V1*T*(Q1**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)
*
*        Compute the generalized singular value decomposition of (A, B)
*           (A = U*D1*(0 R)*(Q**T), B = V*D2*(0 R)*(Q**T))
*
         CALL DTGSJA('U','V','Q',M,P,N,K,L,A,LDA,B,LDB,TOLA,TOLB,ALPHA,
     +               BETA,U,LDU,V,LDV,Q,LDQ,WORK,NCYCLE,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,*)
     +        ' Effective 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)
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Number of cycles of the Kogbetliantz method'
            WRITE (NOUT,99999) NCYCLE
         ELSE
            WRITE (NOUT,99997) 'Failure in DTGSJA. 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,A,I4)
      END
```

9.2 Program Data

```
F08YEF 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

```
 F08YEF Example Program Results

 Number of infinite generalized singular values (K)
     1
 Number of finite generalized singular values (L)
     2
  Effective 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

 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

 Number of cycles of the Kogbetliantz method
     2
```