NAG 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, TOLA,       &
                   TOLB, ALPHA, BETA, U, LDU, V, LDV, Q, LDQ, WORK, NCYCLE,     &
                   INFO)
INTEGER            M, P, N, K, L, LDA, LDB, LDU, LDV, LDQ, NCYCLE, INFO
REAL (KIND=nag_wp) A(LDA,*), B(LDB,*), TOLA, TOLB, ALPHA(N), BETA(N),           &
                   U(LDU,*), V(LDV,*), Q(LDQ,*), WORK(2*N)
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)

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

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

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

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

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

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 nonsingular, and $C^2 + S^2 = 1$. See Section 2.3.5.3 of Anderson *et al.* (1999) 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 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.

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

2: JOBV – CHARACTER(1) *Input*

On entry: if JOBV = 'V', V must contain an orthogonal matrix V_1 on entry, and the product V_1V 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.

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

3: JOBQ – CHARACTER(1) *Input*

On entry: if JOBQ = 'Q', Q must contain an orthogonal matrix Q_1 on entry, and the product Q_1Q 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.

Constraint: JOBQ = 'Q', 'I' 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: 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,$*$) – 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: 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,$*$) – 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: 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: $\mathrm{LDB} \geq \max(1, \mathrm{P})$.

13: TOLA – REAL (KIND=nag_wp) *Input*
14: TOLB – REAL (KIND=nag_wp) *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

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

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

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

On exit: see the description of BETA.

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

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

$\mathrm{ALPHA}(i) = 1$, $\mathrm{BETA}(i) = 0$, for $i = 1, 2, \ldots, k$, and

if $m - k - l \geq 0$, $\mathrm{ALPHA}(i) = \alpha_i$, $\mathrm{BETA}(i) = \beta_i$, for $i = k + 1, \ldots, k + l$, or

if $m - k - l < 0$, $\mathrm{ALPHA}(i) = \alpha_i$, $\mathrm{BETA}(i) = \beta_i$, for $i = k + 1, \ldots, m$ and $\mathrm{ALPHA}(i) = 0$, $\mathrm{BETA}(i) = 1$, for $i = m + 1, \ldots, k + l$.

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

17: U(LDU,$*$) – REAL (KIND=nag_wp) array *Input/Output*

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

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

19: V(LDV,$*$) – REAL (KIND=nag_wp) array *Input/Output*

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

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_1V.

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 $\neq$ 'N', LDV $\geq \max(1, P)$;
otherwise LDV ≥ 1.

21: Q(LDQ,$*$) – REAL (KIND=nag_wp) array *Input/Output*

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

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_1Q.

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

23: WORK($2 \times N$) – REAL (KIND=nag_wp) array *Workspace*

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 $= -i$, argument i 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

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

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 F08YSF (ZTGSJA).

9 Example

This example finds the generalized singular value decomposition

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

of the matrix pair (A, B), where

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

9.1 Program Text

```
    Program f08yefe

!     F08YEF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: dggsvp, dtgsja, 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, j, k, l, lda, &
                                          ldb, ldq, ldu, ldv, m, n, ncycle, p
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:), alpha(:), b(:,:), beta(:),   &
                                          q(:,:), tau(:), u(:,:), v(:,:),      &
                                          work(:)
      Integer, Allocatable             :: iwork(:)
      Character (1)                    :: clabs(1), rlabs(1)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: max, real
!     .. Executable Statements ..
      Write (nout,*) 'F08YEF 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),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 = U1*S*(Q1**T), B = V1*T*(Q1**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)

!     Compute the generalized singular value decomposition of (A, B)
!         (A = U*D1*(0 R)*(Q**T), B = V*D2*(0 R)*(Q**T))
!     The NAG name equivalent of dtgsja is f08yef
      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==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)

        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','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

99999 Format (1X,I5)
99998 Format (3X,8(1P,E12.4))
99997 Format (1X,A,I4)
    End Program f08yefe
```

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
```