

NAG Library Routine Document

F08VGF (DGGSP3)

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

F08VGF (DGGSP3) 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). For sufficiently large problems, a blocked algorithm is used to make best use of level 3 BLAS.

2 Specification

```

SUBROUTINE F08VGF (JOBV, JOBV, JOBQ, M, P, N, A, LDA, B, LDB, TOLA,      &
                  TOLB, K, L, U, LDU, V, LDV, Q, LDQ, IWORK, TAU, WORK,  &
                  LWORK, INFO)
INTEGER             M, P, N, LDA, LDB, K, L, LDU, LDV, LDQ, IWORK(N),  &
                  LWORK, INFO
REAL (KIND=nag_wp) A(LDA,*), B(LDB,*), TOLA, TOLB, U(LDU,*), V(LDV,*), &
                  Q(LDQ,*), TAU(N), WORK(max(1,LWORK))
CHARACTER(1)       JOBV, JOBV, JOBQ

```

The routine may be called by its LAPACK name *dggsvp3*.

3 Description

F08VGF (DGGSP3) computes orthogonal matrices U , V and Q such that

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

$$V^T B Q = \begin{matrix} & \begin{matrix} n-k-l & k & l \end{matrix} \\ l \begin{pmatrix} 0 & 0 & B_{13} \\ 0 & 0 & 0 \end{pmatrix} \\ p-l \end{matrix}$$

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 $(A^T \ B^T)^T$.

This decomposition is usually used as the preprocessing step for computing the Generalized Singular Value Decomposition (GSVD), see routine F08YEF (DTGSJA); the two steps are combined in F08VCF (DGGSP3).

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 (2012) *Matrix Computations* (4th 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: $M \geq 0$.

- 5: P – INTEGER *Input*
On entry: p , the number of rows of the matrix B .
Constraint: $P \geq 0$.

- 6: N – INTEGER *Input*
On entry: n , the number of columns of the matrices A and B .
Constraint: $N \geq 0$.

- 7: A(LDA,*) – REAL (KIND=nag_wp) array *Input/Output*
Note: the second dimension of the array A must be at least $\max(1, N)$.
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 F08VGF (DGGSP3) is called.
Constraint: $LDA \geq \max(1, M)$.

- 9: B(LDB,*) – REAL (KIND=nag_wp) array Input/Output
Note: the second dimension of the array B must be at least $\max(1, N)$.
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 F08VGF (DGGSP3) is called.
Constraint: $LDB \geq \max(1, P)$.
- 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
- $$\begin{aligned} \text{TOLA} &= \max(M, N) \|A\| \epsilon, \\ \text{TOLB} &= \max(P, N) \|B\| \epsilon, \end{aligned}$$
- 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 $(A^T \ B^T)^T$.
- 15: U(LDU,*) – REAL (KIND=nag_wp) array Output
Note: the second dimension of the array U must be at least $\max(1, M)$ if $\text{JOB} = \text{'U'}$, and at least 1 otherwise.
On exit: if $\text{JOB} = \text{'U'}$, U contains the orthogonal matrix U .
 If $\text{JOB} = \text{'N'}$, 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 F08VGF (DGGSP3) is called.
Constraints:
 if $\text{JOB} = \text{'U'}$, $LDU \geq \max(1, M)$;
 otherwise $LDU \geq 1$.
- 17: V(LDV,*) – REAL (KIND=nag_wp) array Output
Note: the second dimension of the array V must be at least $\max(1, P)$ if $\text{JOB} = \text{'V'}$, and at least 1 otherwise.
On exit: if $\text{JOB} = \text{'V'}$, V contains the orthogonal matrix V .
 If $\text{JOB} = \text{'N'}$, 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 F08VGF (DGGSP3) is called.

Constraints:

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

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 F08VGF (DGGSP3) is called.

Constraints:

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

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

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

23: $WORK(\max(1, LWORK))$ – REAL (KIND=nag_wp) array *Workspace*

On exit: if $INFO = 0$, $WORK(1)$ contains the minimum value of $LWORK$ required for optimal performance.

24: $LWORK$ – INTEGER *Input*

On entry: the dimension of the array $WORK$ as declared in the (sub)routin from which F08VGF (DGGSP3) is called.

If $LWORK = -1$, a workspace query is assumed; the routine only calculates the optimal size of the $WORK$ array, returns this value as the first entry of the $WORK$ array, and no error message related to $LWORK$ is issued.

Suggested value: for optimal performance, $LWORK$ must generally be larger than the minimum; increase workspace by, say, $nb \times (N + 1)$, where nb is the optimal **block size**

Constraints:

if $JOBV = 'V'$, $LWORK \geq \max(2 * N + 1, P, M)$;
 if $JOBV = 'N'$, $LWORK \geq \max(2 * N + 1, M)$.

25: $INFO$ – 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.

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 Parallelism and Performance

F08VGF (DGGSP3) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F08VGF (DGGSP3) 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

This routine replaces the deprecated routine F08VEF (DGGSP) which used an unblocked algorithm and therefore did not make best use of level 3 BLAS routines.

The complex analogue of this routine is F08VUF (ZGGSP3).

10 Example

This example finds the generalized factorization

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

of the matrix pair $(A \ B)$, where

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

10.1 Program Text

```

Program f08vgfe

!      F08VGF Example Program Text

!      Mark 26 Release. NAG Copyright 2016.

!      .. Use Statements ..
Use nag_library, Only: dggsvp3, f06raf, f08yef, 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, lwork, m, n,      &
                          ncycle, p
!      .. Local Arrays ..
Real (Kind=nag_wp), Allocatable :: a(:, :), alpha(:), b(:, :), beta(:), &
q(:, :), tau(:), u(:, :), v(:, :), &
work(:)

```

```

Real (Kind=nag_wp)                :: wdum(1)
Integer, Allocatable              :: iwork(:)
Character (1)                    :: clabs(1), rlabs(1)
! .. Intrinsic Procedures ..
Intrinsic                        :: max, nint, real
! .. Executable Statements ..
Write (nout,*) 'F08VGF 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),iwork(n))

! Perform workspace query to get optimal size of work
! The NAG name equivalent of dggsvp3 is f08vgf
lwork = -1
Call dggsvp3('U','V','Q',m,p,n,a,lda,b,ldb,tola,tolb,k,l,u,ldu,v,ldv,q, &
            ldq,iwork,tau,wdum,lwork,info)
lwork = nint(wdum(1))
Allocate (work(lwork))

! 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 dggsvp3 is f08vgf
Call dggsvp3('U','V','Q',m,p,n,a,lda,b,ldb,tola,tolb,k,l,u,ldu,v,ldv,q, &
            ldq,iwork,tau,work,lwork,info)

! Given the factors above find the generalized SVD of (A, B)

! The NAG name equivalent of dtgdja is f08yef
Call f08yef('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)

! Print solution

irank = k + 1
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; B) (k+1)'
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)

```

```

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)

Write (nout,*)
Write (nout,*) 'Number of cycles of the Kogbetliantz method'
Write (nout,99999) ncycle

99999 Format (1X,I5)
99998 Format (3X,8(1P,E12.4))
End Program f08vgfe

```

10.2 Program Data

F08VGF 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

F08VGF Example Program Results

```

Number of infinite generalized singular values (k)
1
Number of finite generalized singular values (l)
2
Effective Numerical rank of (A; B) (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

Number of cycles of the Kogbetliantz method

2
