NAG Fortran Library Routine Document

F08KFF (SORGBR/DORGBR)

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

F08KFF (SORGBR/DORGBR) generates one of the real orthogonal matrices Q or P^T which were determined by F08KEF (SGEBRD/DGEBRD) when reducing a real matrix to bidiagonal form.

2 Specification

```
SUBROUTINE F08KFF(VECT, M, N, K, A, LDA, TAU, WORK, LWORK, INFO)
ENTRY      sorgbr (VECT, M, N, K, A, LDA, TAU, WORK, LWORK, INFO)
INTEGER            M, N, K, LDA, LWORK, INFO
real               A(LDA,*), TAU(*), WORK(*)
CHARACTER*1        VECT
```

The ENTRY statement enables the routine to be called by its LAPACK name.

3 Description

This routine is intended to be used after a call to F08KEF (SGEBRD/DGEBRD), which reduces a real rectangular matrix A to bidiagonal form B by an orthogonal transformation: $A = QBP^T$. F08KEF represents the matrices Q and P^T as products of elementary reflectors.

This routine may be used to generate Q or P^T explicitly as square matrices, or in some cases just the leading columns of Q or the leading rows of P^T.

The various possibilities are specified by the parameters VECT, M, N and K. The appropriate values to cover the most likely cases are as follows (assuming that A was an m by n matrix):

1. To form the full m by m matrix Q:

   ```
   CALL SORGBR ('Q',m,m,n,...)
   ```

   (note that the array A must have at least m columns).

2. If $m > n$, to form the n leading columns of Q:

   ```
   CALL SORGBR ('Q',m,n,n,...)
   ```

3. To form the full n by n matrix P^T:

   ```
   CALL SORGBR ('P',n,n,m,...)
   ```

   (note that the array A must have at least n rows).

4. If $m < n$, to form the m leading rows of P^T:

   ```
   CALL SORGBR ('P',m,n,m,...)
   ```

4 References

Golub G H and van Loan C F (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

5 Parameters

1: VECT – CHARACTER*1 *Input*

On entry: indicates whether the orthogonal matrix Q or P^T is generated as follows:

if VECT = 'Q', Q is generated;

if VECT = 'P', P^T is generated.

Constraint: VECT = 'Q' or 'P'.

2: M – INTEGER *Input*

On entry: the number of rows of the orthogonal matrix Q or P^T to be returned.

Constraint: M $\geq$ 0.

3: N – INTEGER *Input*

On entry: the number of columns of the orthogonal matrix Q or P^T to be returned.

Constraints:

N $\geq$ 0;
if VECT = 'Q', M $\geq$ N $\geq$ K if M > K, or M = N if M $\leq$ K;
if VECT = 'P', N $\geq$ M $\geq$ K if N > K, or N = M if N $\leq$ K.

4: K – INTEGER *Input*

On entry: if VECT = 'Q', the number of columns in the original matrix A; if VECT = 'P', the number of rows in the original matrix A.

Constraint: K $\geq$ 0.

5: A(LDA,*) – ***real*** array *Input/Output*

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

On entry: details of the vectors which define the elementary reflectors, as returned by F08KEF (SGEBRD/DGEBRD).

On exit: the orthogonal matrix Q or P^T, or the leading rows or columns thereof, as specified by VECT, M and N.

6: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08KFF (SORGBR/DORGBR) is called.

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

7: TAU(*) – ***real*** array *Input*

Note: the dimension of the array TAU must be at least $\max(1, \min(\mathrm{M}, \mathrm{K}))$ if VECT = 'Q', and at least $\max(1, \min(\mathrm{N}, \mathrm{K}))$ if VECT = 'P'.

On entry: further details of the elementary reflectors, as returned by F08KEF (SGEBRD/DGEBRD) in its parameter TAUQ if VECT = 'Q', or in its parameter TAUP if VECT = 'P'.

8: WORK(*) – ***real*** array *Workspace*

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

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

9: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08KFF (SORGBR/DORGBR) is called, unless LWORK $= -1$, in which case a workspace query is assumed and the routine only calculates the optimal dimension of WORK (using the formula given below).

Suggested value: for optimum performance LWORK should be at least $\min(\mathrm{M},\mathrm{N}) \times nb$, where nb is the ***blocksize***.

Constraint: LWORK $\geq \max(1, \min(\mathrm{M},\mathrm{N}))$ or LWORK $= -1$.

10: 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.

7 Accuracy

The computed matrix Q differs from an exactly orthogonal matrix by a matrix E such that

$$\|E\|_2 = O(\epsilon),$$

where ϵ is the ***machine precision***. A similar statement holds for the computed matrix P^T.

8 Further Comments

The total number of floating-point operations for the cases listed in Section 3 are approximately as follows:

1. To form the whole of Q:

   $\frac{4}{3}n(3m^2 - 3mn + n^2)$ if $m > n$,

   $\frac{4}{3}m^3$ if $m \leq n$;

2. To form the n leading columns of Q when $m > n$:

   $\frac{2}{3}n^2(3m - n)$;

3. To form the whole of P^T:

   $\frac{4}{3}n^3$ if $m \geq n$,

   $\frac{4}{3}m(3n^2 - 3mn + m^2)$ if $m < n$;

4. To form the m leading rows of P^T when $m < n$:

   $\frac{2}{3}m^2(3n - m)$.

The complex analogue of this routine is F08KTF (CUNGBR/ZUNGBR).

9 Example

For this routine two examples are presented, both of which involve computing the singular value decomposition of a matrix A, where

$$A = \begin{pmatrix} -0.57 & -1.28 & -0.39 & 0.25 \\ -1.93 & 1.08 & -0.31 & -2.14 \\ 2.30 & 0.24 & 0.40 & -0.35 \\ -1.93 & 0.64 & -0.66 & 0.08 \\ 0.15 & 0.30 & 0.15 & -2.13 \\ -0.02 & 1.03 & -1.43 & 0.50 \end{pmatrix}$$

in the first example and

$$A = \begin{pmatrix} -5.42 & 3.28 & -3.68 & 0.27 & 2.06 & 0.46 \\ -1.65 & -3.40 & -3.20 & -1.03 & -4.06 & -0.01 \\ -0.37 & 2.35 & 1.90 & 4.31 & -1.76 & 1.13 \\ -3.15 & -0.11 & 1.99 & -2.70 & 0.26 & 4.50 \end{pmatrix}$$

in the second. A must first be reduced to tridiagonal form by F08KEF (SGEBRD/DGEBRD). The program then calls F08KFF (SORGBR/DORGBR) twice to form Q and P^T, and passes these matrices to F08MEF (SBDSQR/DBDSQR), which computes the singular value decomposition of A.

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.

```
*     F08KFF Example Program Text
*     Mark 16 Release. NAG Copyright 1992.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          MMAX, NMAX, LDA, LDVT, LDU, LDC, LWORK
      PARAMETER        (MMAX=8,NMAX=8,LDA=MMAX,LDVT=NMAX,LDU=MMAX,
     +                 LDC=NMAX,LWORK=64*(MMAX+NMAX))
*     .. Local Scalars ..
      INTEGER          I, IC, IFAIL, INFO, J, M, N
*     .. Local Arrays ..
      real             A(LDA,NMAX), C(LDC,NMAX), D(NMAX), E(NMAX-1),
     +                 TAUP(NMAX), TAUQ(NMAX), U(LDU,NMAX),
     +                 VT(LDVT,NMAX), WORK(LWORK)
*     .. External Subroutines ..
      EXTERNAL         sbdsqr, sgebrd, sorgbr, F06QFF, X04CAF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08KFF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      DO 20 IC = 1, 2
         READ (NIN,*) M, N
         IF (M.LE.MMAX .AND. N.LE.NMAX) THEN
*
*           Read A from data file
*
            READ (NIN,*) ((A(I,J),J=1,N),I=1,M)
*
*           Reduce A to bidiagonal form
*
            CALL sgebrd(M,N,A,LDA,D,E,TAUQ,TAUP,WORK,LWORK,INFO)
*
            IF (M.GE.N) THEN
*
*              Copy A to VT and U
*
               CALL F06QFF('Upper',N,N,A,LDA,VT,LDVT)
*
               CALL F06QFF('Lower',M,N,A,LDA,U,LDU)
*
*              Form P**T explicitly, storing the result in VT
*
```

```
            CALL sorgbr('P',N,N,M,VT,LDVT,TAUP,WORK,LWORK,INFO)
*
*           Form Q explicitly, storing the result in U
*
            CALL sorgbr('Q',M,N,N,U,LDU,TAUQ,WORK,LWORK,INFO)
*
*           Compute the SVD of A
*
            CALL sbdsqr('Upper',N,N,M,0,D,E,VT,LDVT,U,LDU,C,LDC,WORK,
     +                  INFO)
*
*           Print singular values, left & right singular vectors
*
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Example 1: singular values'
            WRITE (NOUT,99999) (D(I),I=1,N)
            WRITE (NOUT,*)
            IFAIL = 0
*
            CALL X04CAF('General',' ',N,N,VT,LDVT,
     +                  'Example 1: right singular vectors, by row',
     +                  IFAIL)
*
            WRITE (NOUT,*)
*
            CALL X04CAF('General',' ',M,N,U,LDU,
     +                  'Example 1: left singular vectors, by column'
     +                  ,IFAIL)
*
         ELSE
*
*           Copy A to VT and U
*
            CALL F06QFF('Upper',M,N,A,LDA,VT,LDVT)
*
            CALL F06QFF('Lower',M,M,A,LDA,U,LDU)
*
*           Form P**T explicitly, storing the result in VT
*
            CALL sorgbr('P',M,N,M,VT,LDVT,TAUP,WORK,LWORK,INFO)
*
*           Form Q explicitly, storing the result in U
*
            CALL sorgbr('Q',M,M,N,U,LDU,TAUQ,WORK,LWORK,INFO)
*
*           Compute the SVD of A
*
            CALL sbdsqr('Lower',M,N,M,0,D,E,VT,LDVT,U,LDU,C,LDC,WORK,
     +                  INFO)
*
*           Print singular values, left & right singular vectors
*
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Example 2: singular values'
            WRITE (NOUT,99999) (D(I),I=1,M)
            WRITE (NOUT,*)
            IFAIL = 0
*
            CALL X04CAF('General',' ',M,N,VT,LDVT,
     +                  'Example 2: right singular vectors, by row',
     +                  IFAIL)
*
            WRITE (NOUT,*)
*
            CALL X04CAF('General',' ',M,M,U,LDU,
     +                  'Example 2: left singular vectors, by column'
     +                  ,IFAIL)
*
         END IF
      END IF
   20 CONTINUE
```

```
      STOP
*
99999 FORMAT (3X,(8F8.4))
      END
```

9.2 Program Data

```
F08KFF Example Program Data
  6  4                                          :Values of M and N, Example 1
 -0.57  -1.28  -0.39   0.25
 -1.93   1.08  -0.31  -2.14
  2.30   0.24   0.40  -0.35
 -1.93   0.64  -0.66   0.08
  0.15   0.30   0.15  -2.13
 -0.02   1.03  -1.43   0.50                     :End of matrix A
  4  6                                          :Values of M and N, Example 2
 -5.42   3.28  -3.68   0.27   2.06   0.46
 -1.65  -3.40  -3.20  -1.03  -4.06  -0.01
 -0.37   2.35   1.90   4.31  -1.76   1.13
 -3.15  -0.11   1.99  -2.70   0.26   4.50   :End of matrix A
```

9.3 Program Results

```
 F08KFF Example Program Results

 Example 1: singular values
     3.9987  3.0005  1.9967  0.9999

 Example 1: right singular vectors, by row
          1       2       3       4
 1   0.8251 -0.2794  0.2048  0.4463
 2  -0.4530 -0.2121 -0.2622  0.8252
 3  -0.2829 -0.7961  0.4952 -0.2026
 4   0.1841 -0.4931 -0.8026 -0.2807

 Example 1: left singular vectors, by column
          1       2       3       4
 1  -0.0203  0.2794  0.4690  0.7692
 2  -0.7284 -0.3464 -0.0169 -0.0383
 3   0.4393 -0.4955 -0.2868  0.0822
 4  -0.4678  0.3258 -0.1536 -0.1636
 5  -0.2200 -0.6428  0.1125  0.3572
 6  -0.0935  0.1927 -0.8132  0.4957

 Example 2: singular values
     7.9987  7.0059  5.9952  4.9989

 Example 2: right singular vectors, by row
          1       2       3       4       5       6
 1  -0.7933  0.3163 -0.3342 -0.1514  0.2142  0.3001
 2   0.1002  0.6442  0.4371  0.4890  0.3771  0.0501
 3   0.0111  0.1724 -0.6367  0.4354 -0.0430 -0.6111
 4   0.2361  0.0216 -0.1025 -0.5286  0.7460 -0.3120

 Example 2: left singular vectors, by column
          1       2       3       4
 1   0.8884  0.1275  0.4331  0.0838
 2   0.0733 -0.8264  0.1943 -0.5234
 3  -0.0361  0.5435  0.0756 -0.8352
 4   0.4518 -0.0733 -0.8769 -0.1466
```