NAG Fortran Library Routine Document

F02XUF

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

F02XUF returns all, or part, of the singular value decomposition of a complex upper triangular matrix.

2 Specification

```
      SUBROUTINE F02XUF(N, A, LDA, NCOLB, B, LDB, WANTQ, Q, LDQ, SV, WANTP,
     1                  RWORK, CWORK, IFAIL)
      INTEGER           N, LDA, NCOLB, LDB, LDQ, IFAIL
      real              SV(*), RWORK(*)
      complex           A(LDA,*), B(LDB,*), Q(LDQ,*), CWORK(*)
      LOGICAL           WANTQ, WANTP
```

3 Description

The n by n upper triangular matrix R is factorized as

$$R = QSP^H,$$

where Q and P are n by n unitary matrices and S is an n by n diagonal matrix with real non-negative diagonal elements, $sv_1, sv_2, \ldots, sv_n$, ordered such that

$$sv_1 \geq sv_2 \geq \cdots \geq sv_n \geq 0.$$

The columns of Q are the left-hand singular vectors of R, the diagonal elements of S are the singular values of R and the columns of P are the right-hand singular vectors of R.

Either or both of Q and P^H may be requested and the matrix C given by

$$C = Q^H B,$$

where B is an n by $ncolb$ given matrix, may also be requested.

The routine obtains the singular value decomposition by first reducing R to bidiagonal form by means of Givens plane rotations and then using the QR algorithm to obtain the singular value decomposition of the bidiagonal form.

Good background descriptions to the singular value decomposition are given in Dongarra *et al.* (1979), Hammarling (1985) and Wilkinson (1978).

Note that if K is any unitary diagonal matrix so that

$$KK^H = I,$$

then

$$A = (QK)S(PK)^H$$

is also a singular value decomposition of A.

4 References

Dongarra J J, Moler C B, Bunch J R and Stewart G W (1979) *LINPACK Users' Guide* SIAM, Philadelphia

Hammarling S (1985) The singular value decomposition in multivariate statistics *SIGNUM Newsl.* **20 (3)** 2–25

Wilkinson J H (1978) Singular Value Decomposition – Basic Aspects *Numerical Software – Needs and Availability* (ed D A H Jacobs) Academic Press

5 Parameters

1:    N – INTEGER   *Input*

On entry: the order, n, of the matrix R.

When N = 0 then an immediate return is effected.

Constraint: N $\geq$ 0.

2:    A(LDA,*) – **complex** array   *Input*

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

On entry: the leading n by n upper triangular part of the array A must contain the upper triangular matrix R.

On exit: if WANTP = .TRUE., then the n by n part of A will contain the n by n unitary matrix P^H, otherwise the n by n upper triangular part of A is used as internal workspace, but the strictly lower triangular part of A is not referenced.

3:    LDA – INTEGER   *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F02XUF is called.

Constraint: LDA $\geq$ max$(1, N)$.

4:    NCOLB – INTEGER   *Input*

On entry: the number of columns, $ncolb$, of the matrix B.

When NCOLB = 0 the array B is not referenced.

Constraint: NCOLB $\geq$ 0.

5:    B(LDB,*) – **complex** array   *Input/Output*

Note: the second dimension of the array B must be at least max$(1, NCOLB)$.

On entry: if NCOLB > 0, the leading n by $ncolb$ part of the array B must contain the matrix to be transformed.

On exit: B is overwritten by the n by $ncolb$ matrix $Q^H B$.

6:    LDB – INTEGER   *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F02XUF is called.

Constraint: if NCOLB > 0 then LDB $\geq$ max$(1, N)$.

7:    WANTQ – LOGICAL   *Input*

On entry: WANTQ must be .TRUE. if the matrix Q is required. If WANTQ = .FALSE. then the array Q is not referenced.

8:    Q(LDQ,*) – **complex** array   *Output*

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

On exit: if WANTQ = .TRUE., the leading n by n part of the array Q will contain the unitary matrix Q. Otherwise the array Q is not referenced.

9: LDQ – INTEGER *Input*

On entry: the first dimension of the array Q as declared in the (sub)program from which F02XUF is called.

Constraint: if WANTQ = .TRUE., LDQ $\geq$ max$(1, N)$.

10: SV(∗) – ***real*** array *Output*

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

On exit: the n diagonal elements of the matrix S.

11: WANTP – LOGICAL *Input*

On entry: WANTP must be .TRUE. if the matrix P^H is required, in which case P^H is returned in the array A, otherwise WANTP must be .FALSE..

12: RWORK(∗) – ***real*** array *Output*

Note: the dimension of the array RWORK must be at least $lrwork$, where $lrwork$ must satisfy:

$$lrwork = 2 \times (N - 1)$$

when NCOLB = 0 and WANTQ and WANTP are .FALSE.,

$$lrwork = 3 \times (N - 1)$$

when either NCOLB = 0 and WANTQ = .FALSE. and WANTP = .TRUE., or WANTP = .FALSE. and one or both of NCOLB > 0 and WANTQ = .TRUE.

$$lrwork = 5 \times (N - 1)$$

otherwise.

On exit: RWORK}(N) contains the total number of iterations taken by the QR algorithm.

The rest of the array is used as workspace.

13: CWORK(∗) – ***complex*** array *Workspace*

Note: the length of CWORK must be at least max$(1, n - 1)$.

14: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

On exit: IFAIL = 0 unless the routine detects an error (see Section 6).

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, for users not familiar with this parameter the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

6 Error Indicators and Warnings

If on entry IFAIL = 0 or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL = -1

On entry, N < 0,
or   LDA < N,
or   NCOLB < 0,

or	LDB < N and NCOLB > 0,
or	LDQ < N and WANTQ = .TRUE.

IFAIL > 0

The QR algorithm has failed to converge in $50 \times$ N iterations. In this case $\mathrm{SV}(1), \mathrm{SV}(2), \ldots, \mathrm{SV}(\mathrm{IFAIL})$ may not have been found correctly and the remaining singular values may not be the smallest. The matrix R will nevertheless have been factorized as $R = QEP^H$, where E is a bidiagonal matrix with $\mathrm{SV}(1), \mathrm{SV}(2), \ldots, \mathrm{SV}(n)$ as the diagonal elements and $\mathrm{RWORK}(1), \mathrm{RWORK}(2), \ldots, \mathrm{RWORK}(n-1)$ as the super-diagonal elements.

This failure is not likely to occur.

7 Accuracy

The computed factors Q, S and P satisfy the relation

$$QSP^H = A + E,$$

where

$$\|E\| \leq c\epsilon\|A\|,$$

ϵ is the **machine precision**, c is a modest function of n and $\|.\|$ denotes the spectral (two) norm. Note that $\|A\| = sv_1$.

8 Further Comments

For given values of NCOLB, WANTQ and WANTP, the number of floating-point operations required is approximately proportional to n^3.

Following the use of this routine the rank of R may be estimated by a call to the INTEGER FUNCTION F06KLF. The statement

```
IRANK = F06KLF(N,SV,1,TOL)
```

returns the value $(k-1)$ in IRANK, where k is the smallest integer for which $sv(k) < tol \times sv(1)$, where tol is the tolerance supplied in TOL, so that IRANK is an estimate of the rank of S and thus also of R. If TOL is supplied as negative then the **machine precision** is used in place of TOL.

9 Example

To find the singular value decomposition of the 3 by 3 upper triangular matrix

$$A = \begin{pmatrix} 1 & 1+i & 1+i \\ 0 & -2 & -1-i \\ 0 & 0 & -3 \end{pmatrix}$$

together with the vector $Q^H b$ for the vector

$$b = \begin{pmatrix} 1+1i \\ -1 \\ -1+1i \end{pmatrix}.$$

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.

```
*     F02XUF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
```

```
      INTEGER           NMAX, NCOLB
      PARAMETER         (NMAX=10,NCOLB=1)
      INTEGER           LDA, LDB, LDQ
      PARAMETER         (LDA=NMAX,LDB=NMAX,LDQ=NMAX)
      INTEGER           LRWORK
      PARAMETER         (LRWORK=5*(NMAX-1))
      INTEGER           LCWORK
      PARAMETER         (LCWORK=NMAX-1)
*     .. Local Scalars ..
      INTEGER           I, IFAIL, J, N
      LOGICAL           WANTP, WANTQ
*     .. Local Arrays ..
      complex           A(LDA,NMAX), B(LDB), CWORK(LCWORK), Q(LDQ,NMAX)
      real              RWORK(LRWORK), SV(NMAX)
*     .. External Subroutines ..
      EXTERNAL          F02XUF
*     .. Intrinsic Functions ..
      INTRINSIC         conjg
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F02XUF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      WRITE (NOUT,*)
      IF (N.GT.NMAX) THEN
         WRITE (NOUT,*) 'N is out of range.'
         WRITE (NOUT,99999) 'N = ', N
      ELSE
         READ (NIN,*) ((A(I,J),J=I,N),I=1,N)
         READ (NIN,*) (B(I),I=1,N)
*     Find the SVD of A.
         WANTQ = .TRUE.
         WANTP = .TRUE.
         IFAIL = 0
*
         CALL F02XUF(N,A,LDA,NCOLB,B,LDB,WANTQ,Q,LDQ,SV,WANTP,RWORK,
     +               CWORK,IFAIL)
*
         WRITE (NOUT,*) 'Singular value decomposition of A'
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Singular values'
         WRITE (NOUT,99998) (SV(I),I=1,N)
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Left-hand singular vectors, by column'
         DO 20 I = 1, N
            WRITE (NOUT,99997) (Q(I,J),J=1,N)
   20    CONTINUE
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Right-hand singular vectors, by column'
         DO 40 I = 1, N
            WRITE (NOUT,99997) (conjg(A(J,I)),J=1,N)
   40    CONTINUE
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Vector conjg( Q'' )*B'
         WRITE (NOUT,99997) (B(I),I=1,N)
      END IF
      STOP
*
99999 FORMAT (1X,A,I5)
99998 FORMAT (1X,3F9.4)
99997 FORMAT (1X,3('(',F7.4,',',F8.4,') ',:))
      END
```

9.2   Program Data

```
F02XUF Example Program Data

  3                                            :Value of N

  (1.0,  0.0)   ( 1.0,  1.0)   ( 1.0,  1.0)
                (-2.0,  0.0)   (-1.0, -1.0)
                               (-3.0,  0.0)    :End of matrix A

  (1.0,  1.0)   (-1.0,  0.0)   (-1.0,  1.0)    :End of vector B
```

9.3   Program Results

```
F02XUF Example Program Results

Singular value decomposition of A

Singular values
    3.9263    2.0000    0.7641

Left-hand singular vectors, by column
(-0.5005, -0.0000) (-0.4529,  0.0000) ( 0.7378,  0.0000)
( 0.5152, -0.1514) ( 0.1132, -0.5661) ( 0.4190, -0.4502)
( 0.4041, -0.5457) (-0.0000,  0.6794) ( 0.2741,  0.0468)

Right-hand singular vectors, by column
(-0.1275, -0.0000) (-0.2265, -0.0000) ( 0.9656, -0.0000)
(-0.3899,  0.2046) (-0.3397,  0.7926) (-0.1311,  0.2129)
(-0.5289,  0.7142) ( 0.0000, -0.4529) (-0.0698, -0.0119)

Vector conjg( Q' )*B
(-1.9656, -0.7935) ( 0.1132, -0.3397) ( 0.0915,  0.6086)
```