NAG Fortran Library Routine Document

F02XEF

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

F02XEF returns all, or part, of the singular value decomposition of a general complex matrix.

2   Specification

```
     SUBROUTINE F02XEF (M, N, A, LDA, NCOLB, B, LDB, WANTQ, Q, LDQ, SV,
    1                   WANTP, PH, LDPH, RWORK, CWORK, IFAIL)
     INTEGER           M, N, LDA, NCOLB, LDB, LDQ, LDPH, IFAIL
     double precision  SV(*), RWORK(*)
     complex*16        A(LDA,*), B(LDB,*), Q(LDQ,*), PH(LDPH,*), CWORK(*)
     LOGICAL           WANTQ, WANTP
```

3   Description

The m by n matrix A is factorized as

$$A = QDP^{\mathrm{H}},$$

where

$$D = \begin{pmatrix} S \\ 0 \end{pmatrix}, \qquad m > n,$$

$$D = S, \qquad m = n,$$

$$D = \begin{pmatrix} S & 0 \end{pmatrix}, \quad m < n,$$

Q is an m by m unitary matrix, P is an n by n unitary matrix and S is a $\min(m,n)$ by $\min(m,n)$ diagonal matrix with real non-negative diagonal elements, $sv_1, sv_2, \ldots, sv_{\min(m,n)}$, ordered such that

$$sv_1 \geq sv_2 \geq \cdots \geq sv_{\min(m,n)} \geq 0.$$

The first $\min(m,n)$ columns of Q are the left-hand singular vectors of A, the diagonal elements of S are the singular values of A and the first $\min(m,n)$ columns of P are the right-hand singular vectors of A.

Either or both of the left-hand and right-hand singular vectors of A may be requested and the matrix C given by

$$C = Q^{\mathrm{H}}B,$$

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

The routine obtains the singular value decomposition by first reducing A to upper triangular form by means of Householder transformations, from the left when $m \geq n$ and from the right when $m < n$. The upper triangular form is then reduced to bidiagonal form by Givens plane rotations and finally the QR algorithm is used 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 this routine is not based on the LINPACK routine CSVDC/ZSVDC.

Note that if K is any unitary diagonal matrix so that

$$KK^{\mathrm{H}} = I,$$

then

$$A = (QK)D(PK)^{\mathrm{H}}$$

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:    M – INTEGER   *Input*

On entry: m, the number of rows of the matrix A.

Constraint: M $\geq$ 0.

If M $= 0$, an immediate return is effected

2:    N – INTEGER   *Input*

On entry: n, the number of columns of the matrix A.

Constraint: N $\geq$ 0.

If N $= 0$, an immediate return is effected

3:    A(LDA,$*$) – **complex*16** array   *Input/Output*

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

On entry: the leading m by n part of the array A must contain the matrix A whose singular value decomposition is required.

On exit: if M $\geq$ N and WANTQ $=$ .TRUE., the leading m by n part of A will contain the first n columns of the unitary matrix Q.

If M $<$ N and WANTP $=$ .TRUE., the leading m by n part of A will contain the first m rows of the unitary matrix P^{H}.

If M $\geq$ N and WANTQ $=$ .FALSE. and WANTP $=$ .TRUE., the leading n by n part of A will contain the first n rows of the unitary matrix P^{H}.

Otherwise the leading m by n part of A is used as internal workspace.

4:    LDA – INTEGER   *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F02XEF is called.

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

5:    NCOLB – INTEGER   *Input*

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

If NCOLB $= 0$, the array B is not referenced.

Constraint: NCOLB $\geq$ 0.

6: B(LDB,∗) – **complex*16** array *Input/Output*

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

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

On exit: is overwritten by the m by $ncolb$ matrix $Q^{\mathrm{H}}B$.

7: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F02XEF is called.

Constraints:

> if $\text{NCOLB} > 0$, $\text{LDB} \geq \max(1, \text{M})$;
> $\text{LDB} \geq 1$ otherwise.

8: WANTQ – LOGICAL *Input*

On entry: must be .TRUE. if the left-hand singular vectors are required.

If WANTQ = .FALSE., the array Q is not referenced.

9: Q(LDQ,∗) – **complex*16** array *Output*

Note: the second dimension of the array Q must be at least $\max(1, \text{M})$ if $\text{M} < \text{N}$ and WANTQ = .TRUE., and at least 1 otherwise.

On exit: if $\text{M} < \text{N}$ and WANTQ = .TRUE., the leading m by m part of the array Q will contain the unitary matrix Q. Otherwise the array Q is not referenced.

10: LDQ – INTEGER *Input*

On entry: the first dimension of the array Q as declared in the (sub)program from which F02XEF is called.

Constraints:

> if $\text{M} < \text{N}$ and WANTQ = .TRUE., $\text{LDQ} \geq \max(1, \text{M})$;
> $\text{LDQ} \geq 1$ otherwise.

11: SV(∗) – **double precision** array *Output*

Note: the dimension of the array SV must be at least $\min(\text{M}, \text{N})$.

On exit: the $\min(m, n)$ diagonal elements of the matrix S.

12: WANTP – LOGICAL *Input*

On entry: must be .TRUE. if the right-hand singular vectors are required.

If WANTP = .FALSE., the array PH is not referenced.

13: PH(LDPH,∗) – **complex*16** array *Output*

Note: the second dimension of the array PH must be at least $\max(1, \text{N})$ if $\text{M} \geq \text{N}$ and WANTQ = .TRUE. and WANTP = .TRUE., and at least 1 otherwise.

On exit: if $\text{M} \geq \text{N}$ and WANTQ and WANTP are .TRUE., the leading n by n part of the array PH will contain the unitary matrix P^{H}. Otherwise the array PH is not referenced.

14: LDPH – INTEGER *Input*

On entry: the first dimension of the array PH as declared in the (sub)program from which F02XEF is called.

Constraints:

> if $M \geq N$ and WANTQ = .TRUE. and WANTP = .TRUE., $LDPH \geq \max(1, N)$;
> $LDPH \geq 1$ otherwise.

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

Note: the dimension of the array RWORK must be at least $\max(1, lrwork)$, where $lrwork$ must satisfy:

> $lrwork = 2 \times (\min(M, N) - 1)$, when NCOLB = 0 and WANTQ and WANTP are .FALSE.,
>
> $lrwork = 3 \times (\min(M, N) - 1)$, when either NCOLB = 0 and WANTQ = .FALSE. and WANTP = .TRUE., or WANTP = .FALSE. and one or both of NCOLB > 0 and WANTQ = .TRUE.,
>
> $lrwork = 5 \times (\min(M, N) - 1)$, otherwise.

On exit: $\text{RWORK}(\min(M, N))$ contains the total number of iterations taken by the *QR* algorithm.

The rest of the array is used as workspace.

16: CWORK($*$) – **complex*16** array *Workspace*

Note: the dimension of the array CWORK must be at least $\max(1, lcwork)$, where $lcwork$ must satisfy:

> $lcwork = N + \max(N^2, NCOLB)$, when $M \geq N$ and WANTQ and WANTP are both .TRUE.,
>
> $lcwork = N + \max(N^2 + N, NCOLB)$, when $M \geq N$ and WANTQ = .TRUE., but WANTP = .FALSE.,
>
> $lcwork = N + \max(N, NCOLB)$, when $M \geq N$ and WANTQ = .FALSE.,
>
> $lcwork = M^2 + M + \max(1, M - 1)$, when $M < N$ and WANTP = .TRUE.,
>
> $lcwork = M$, when $M < N$ and WANTP = .FALSE..

17: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you 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, if you are 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

> One or more of the following conditions hold:
>
> > M < 0;
> > N < 0;
> > LDA < M;
> > NCOLB < 0;
> > LDB < M and NCOLB > 0;

LDQ $<$ M and M $<$ N and WANTQ $=$ .TRUE.;
LDPH $<$ N and M $\geq$ N and WANTQ $=$ .TRUE. and WANTP $=$ .TRUE..

IFAIL > 0

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

This failure is not likely to occur.

7 Accuracy

The computed factors Q, D and P satisfy the relation

$$QDP^{\text{H}} = A + E,$$

where

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

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

8 Further Comments

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

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

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

9 Example

For this routine two examples are presented. There is a single example program for F02XEF, with a main program and the code to solve the two example problems is given in the (sub)programs EX1 and EX2.

Example 1 (EX1)

To find the singular value decomposition of the 5 by 3 matrix

$$A = \begin{pmatrix} 0.5i & -0.5 + 1.5i & -1.0 + 1.0i \\ 0.4 + 0.3i & 0.9 + 1.3i & 0.2 + 1.4i \\ 0.4 & -0.4 + 0.4i & 1.8 \\ 0.3 - 0.4i & 0.1 + 0.7i & 0.0 \\ -0.3i & 0.3 + 0.3i & 2.4i \end{pmatrix},$$

together with the vector $Q^{\text{H}}b$ for the vector

$$b = \begin{pmatrix} -0.55 + 1.05i \\ 0.49 + 0.93i \\ 0.56 - 0.16i \\ 0.39 + 0.23i \\ 1.13 + 0.83i \end{pmatrix}.$$

Example 2 (EX2)

To find the singular value decomposition of the 3 by 5 matrix

$$A = \begin{pmatrix} 0.5i & 0.4 - 0.3i & 0.4 & 0.3 + 0.4i & 0.3i \\ -0.5 - 1.5i & 0.9 - 1.3i & -0.4 - 0.4i & 0.1 - 0.7i & 0.3 - 0.3i \\ -1.0 - 1.0i & 0.2 - 1.4i & 1.8 & 0.0 & -2.4i \end{pmatrix}.$$

9.1   Program Text

```
*      F02XEF Example Program Text
*      Mark 17 Revised. NAG Copyright 1995.
*      .. Parameters ..
       INTEGER           NOUT
       PARAMETER         (NOUT=6)
*      .. External Subroutines ..
       EXTERNAL          EX1, EX2
*      .. Executable Statements ..
       WRITE (NOUT,*) 'F02XEF Example Program Results'
       CALL EX1
       CALL EX2
       STOP
       END
*
       SUBROUTINE EX1
*      .. Parameters ..
       INTEGER           NIN, NOUT
       PARAMETER         (NIN=5,NOUT=6)
       INTEGER           MMAX, NMAX, NCOLB
       PARAMETER         (MMAX=5,NMAX=3,NCOLB=1)
       INTEGER           LDA, LDB, LDPH
       PARAMETER         (LDA=MMAX,LDB=MMAX,LDPH=NMAX)
       INTEGER           LRWORK
       PARAMETER         (LRWORK=5*(NMAX-1))
       INTEGER           LCWORK
       PARAMETER         (LCWORK=NMAX**2+NMAX)
*      .. Local Scalars ..
       INTEGER           I, IFAIL, J, M, N
       LOGICAL           WANTP, WANTQ
*      .. Local Arrays ..
       COMPLEX *16       A(LDA,NMAX), B(LDB), CWORK(LCWORK), DUMMY(1),
      +                  PH(LDPH,NMAX)
       DOUBLE PRECISION RWORK(LRWORK), SV(NMAX)
*      .. External Subroutines ..
       EXTERNAL          F02XEF
*      .. Intrinsic Functions ..
       INTRINSIC         CONJG
*      .. Executable Statements ..
       WRITE (NOUT,*)
       WRITE (NOUT,*)
       WRITE (NOUT,*) 'Example 1'
*      Skip heading in data file
       READ (NIN,*)
       READ (NIN,*)
       READ (NIN,*)
       READ (NIN,*) M, N
       WRITE (NOUT,*)
       IF ((M.GT.MMAX) .OR. (N.GT.NMAX)) THEN
          WRITE (NOUT,*) 'M or N is out of range.'
          WRITE (NOUT,99999) 'M = ', M, '   N = ', N
       ELSE
          READ (NIN,*) ((A(I,J),J=1,N),I=1,M)
          READ (NIN,*) (B(I),I=1,M)
*      Find the SVD of A.
          WANTQ = .TRUE.
          WANTP = .TRUE.
          IFAIL = 0
*
          CALL F02XEF(M,N,A,LDA,NCOLB,B,LDB,WANTQ,DUMMY,1,SV,WANTP,PH,
      +               LDPH,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, M
            WRITE (NOUT,99997) (A(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(PH(J,I)),J=1,N)
   40    CONTINUE
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Vector conjg( Q'' )*B'
         WRITE (NOUT,99997) (B(I),I=1,M)
      END IF
*
99999 FORMAT (1X,A,I5,A,I5)
99998 FORMAT (1X,5F9.4)
99997 FORMAT ((1X,3('(',F7.4,',',F8.4,') ',:)))
      END
*
      SUBROUTINE EX2
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          MMAX, NMAX
      PARAMETER        (MMAX=3,NMAX=5)
      INTEGER          LDA, LDQ
      PARAMETER        (LDA=MMAX,LDQ=MMAX)
      INTEGER          LRWORK
      PARAMETER        (LRWORK=5*(MMAX-1))
      INTEGER          LCWORK
      PARAMETER        (LCWORK=MMAX**2+2*MMAX-1)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, J, M, N, NCOLB
      LOGICAL          WANTP, WANTQ
*     .. Local Arrays ..
      COMPLEX *16      A(LDA,NMAX), CWORK(LCWORK), DUMMY(1), Q(LDQ,MMAX)
      DOUBLE PRECISION RWORK(LRWORK), SV(MMAX)
*     .. External Subroutines ..
      EXTERNAL         F02XEF
*     .. Intrinsic Functions ..
      INTRINSIC        CONJG
*     .. Executable Statements ..
      WRITE (NOUT,*)
      WRITE (NOUT,*)
      WRITE (NOUT,*) 'Example 2'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*)
      READ (NIN,*) M, N
      WRITE (NOUT,*)
      IF ((M.GT.MMAX) .OR. (N.GT.NMAX)) THEN
         WRITE (NOUT,*) 'M or N is out of range.'
         WRITE (NOUT,99999) 'M = ', M, '   N = ', N
      ELSE
         READ (NIN,*) ((A(I,J),J=1,N),I=1,M)
*        Find the SVD of A.
         WANTQ = .TRUE.
         WANTP = .TRUE.
         NCOLB = 0
         IFAIL = 0
*
         CALL F02XEF(M,N,A,LDA,NCOLB,DUMMY,1,WANTQ,Q,LDQ,SV,WANTP,DUMMY,
     +              1,RWORK,CWORK,IFAIL)
*
         WRITE (NOUT,*) 'Singular value decomposition of A'
```

```
          WRITE (NOUT,*)
          WRITE (NOUT,*) 'Singular values'
          WRITE (NOUT,99998) (SV(I),I=1,M)
          WRITE (NOUT,*)
          WRITE (NOUT,*) 'Left-hand singular vectors, by column'
          DO 20 I = 1, M
             WRITE (NOUT,99997) (Q(I,J),J=1,M)
   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,M)
   40     CONTINUE
       END IF
*
99999 FORMAT (1X,A,I5,A,I5)
99998 FORMAT (1X,5F9.4)
99997 FORMAT (1X,3('(',F7.4,',',F8.4,') ',:))
      END
```

9.2 Program Data

```
F02XEF Example Program Data

Example 1
  5    3                                  :Values of M and N

( 0.00, 0.50) (-0.50, 1.50) (-1.00, 1.00)
( 0.40, 0.30) ( 0.90, 1.30) ( 0.20, 1.40)
( 0.40, 0.00) (-0.40, 0.40) ( 1.80, 0.00)
( 0.30,-0.40) ( 0.10, 0.70) ( 0.00, 0.00)
( 0.00,-0.30) ( 0.30, 0.30) ( 0.00, 2.40)   :End of matrix A

(-0.55, 1.05) ( 0.49, 0.93) ( 0.56,-0.16)
( 0.39, 0.23) ( 1.13, 0.83)                 :End of vector B

Example 2
  3    5                                  :Values of M and N

( 0.00,-0.50) ( 0.40,-0.30) ( 0.40, 0.00) ( 0.30, 0.40) ( 0.00, 0.30)
(-0.50,-1.50) ( 0.90,-1.30) (-0.40,-0.40) ( 0.10,-0.70) ( 0.30,-0.30)
(-1.00,-1.00) ( 0.20,-1.40) ( 1.80, 0.00) ( 0.00, 0.00) ( 0.00,-2.40)
                                            :End of matrix A
```

9.3 Program Results

```
F02XEF Example Program Results


Example 1

Singular value decomposition of A

Singular values
   3.9263   2.0000   0.7641

Left-hand singular vectors, by column
(-0.0757, -0.5079) (-0.2831, -0.2831) (-0.2251,  0.1594)
(-0.4517, -0.2441) (-0.3963,  0.0566) (-0.0075,  0.2757)
(-0.2366,  0.2669) (-0.1359, -0.6341) ( 0.2983, -0.2082)
(-0.0561, -0.0513) (-0.3284, -0.0340) ( 0.1670, -0.5978)
(-0.4820, -0.3277) ( 0.3737,  0.1019) (-0.0976, -0.5664)

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)
```

```
(-0.0600, -0.0200) ( 0.0400,  0.1200)


Example 2

Singular value decomposition of A

Singular values
   3.9263   2.0000   0.7641

Left-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)

Right-hand singular vectors, by column
(-0.0757, -0.5079) ( 0.2831,  0.2831) ( 0.2251, -0.1594)
(-0.4517, -0.2441) ( 0.3963, -0.0566) ( 0.0075, -0.2757)
(-0.2366,  0.2669) ( 0.1359,  0.6341) (-0.2983,  0.2082)
(-0.0561, -0.0513) ( 0.3284,  0.0340) (-0.1670,  0.5978)
(-0.4820, -0.3277) (-0.3737, -0.1019) ( 0.0976,  0.5664)
```