F02WEF – NAG Fortran Library Routine Document

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

F02WEF returns all, or part, of the singular value decomposition of a general real matrix.

2 Specification

```
 SUBROUTINE F02WEF(M, N, A, LDA, NCOLB, B, LDB, WANTQ, Q, LDQ, SV,
1                  WANTP, PT, LDPT, WORK, IFAIL)
 INTEGER           M, N, LDA, NCOLB, LDB, LDQ, LDPT, IFAIL
 real              A(LDA,*), B(LDB,*), Q(LDQ,*), SV(*), PT(LDPT,*),
1                  WORK(*)
 LOGICAL           WANTQ, WANTP
```

3 Description

The m by n matrix A is factorized as

$$A = QDP^T,$$

where

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

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

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

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

$$C = Q^T B,$$

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.* [1], Hammarling [2] and Wilkinson [3]. Note that this routine is not based on the LINPACK routine SSVDC/DSVDC.

Note that if K is any orthogonal diagonal matrix so that

$$KK^T = I$$

(so that K has elements $+1$ or -1 on the diagonal), then

$$A = (QK)D(PK)^T$$

is also a singular value decomposition of A.

4 References

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

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

[3] 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: the number of rows, m, of the matrix A.

Constraint: $\mathrm{M} \geq 0$.

When $\mathrm{M} = 0$ then an immediate return is effected.

2: N — INTEGER *Input*

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

Constraint: $\mathrm{N} \geq 0$.

When $\mathrm{N} = 0$ then an immediate return is effected.

3: A(LDA,∗) — ***real*** array *Input/Output*

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

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 $\mathrm{M} \geq \mathrm{N}$ and WANTQ = .TRUE., then the leading m by n part of A will contain the first n columns of the orthogonal matrix Q.

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

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

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 F02WEF is called.

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

5: NCOLB — INTEGER *Input*

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

When $\mathrm{NCOLB} = 0$ the array B is not referenced.

Constraint: $\mathrm{NCOLB} \geq 0$.

6: B(LDB,∗) — ***real*** array *Input/Output*

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

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

On exit: B is overwritten by the m by *ncolb* matrix $Q^T B$.

7: LDB — INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F02WEF is called.

Constraint: if NCOLB > 0 then LDB $\geq \max(1, \mathrm{M})$.

8: WANTQ — LOGICAL *Input*

On entry: WANTQ must be .TRUE., if the left-hand singular vectors are required. If WANTQ = .FALSE., then the array Q is not referenced.

9: Q(LDQ,∗) — ***real*** array *Output*

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

On exit: if M $<$ N and WANTQ = .TRUE., the leading m by m part of the array Q will contain the orthogonal 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 F02WEF is called.

Constraint: if M $<$ N and WANTQ = .TRUE., LDQ $\geq \max(1, \mathrm{M})$.

11: SV(∗) — ***real*** array *Output*

Note. The length of SV must be at least $\min(\mathrm{M}, \mathrm{N})$.

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

12: WANTP — LOGICAL *Input*

On entry: WANTP must be .TRUE. if the right-hand singular vectors are required. If WANTP = .FALSE., then the array PT is not referenced.

13: PT(LDPT,∗) — ***real*** array *Output*

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

On exit: if M $\geq$ N and WANTQ and WANTP are .TRUE., the leading n by n part of the array PT will contain the orthogonal matrix P^T. Otherwise the array PT is not referenced.

14: LDPT — INTEGER *Input*

On entry: the first dimension of the array PT as declared in the (sub)program from which F02WEF is called.

Constraint: if M $\geq$ N and WANTQ and WANTP are .TRUE., LDPT $\geq \max(1, \mathrm{N})$.

15: WORK(∗) — ***real*** array *Output*

Note. The length of WORK must be at least $\max(1, lwork)$, where $lwork$ must be as given in the following table:

M $\geq$ N WANTQ = .TRUE. and WANTP = .TRUE.

$$lwork = \max(\mathrm{N}^2 + 5 \times (\mathrm{N}-1), \mathrm{N}+\mathrm{NCOLB}, 4)$$

WANTQ = .TRUE. and WANTP = .FALSE.

$$lwork = \max(\mathrm{N}^2 + 4 \times (\mathrm{N}-1), \mathrm{N}+\mathrm{NCOLB}, 4)$$

WANTQ = .FALSE. and WANTP = .TRUE.

$$lwork = \max(3 \times (\mathrm{N}-1), 2) \quad \text{when NCOLB} = 0$$

$$lwork = \max(5 \times (\mathrm{N}-1), 2) \quad \text{when NCOLB} > 0$$

WANTQ = .FALSE. and WANTP = .FALSE.

$$lwork = \max(2 \times (\mathrm{N}-1), 2) \quad \text{when NCOLB} = 0$$

$$lwork = \max(3 \times (\mathrm{N}-1), 2) \quad \text{when NCOLB} > 0$$

M < N WANTQ = .TRUE. and WANTP = .TRUE.

$$lwork = \max(\mathrm{M}^2 + 5 \times (\mathrm{M} - 1), 2)$$

WANTQ = .TRUE. and WANTP = .FALSE.

$$lwork = \max(3 \times (\mathrm{M} - 1), 1)$$

WANTQ = .FALSE. and WANTP = .TRUE.

$$lwork = \max(\mathrm{M}^2 + 3 \times (\mathrm{M} - 1), 2) \quad \text{when NCOLB} = 0$$

$$lwork = \max(\mathrm{M}^2 + 5 \times (\mathrm{M} - 1), 2) \quad \text{when NCOLB} > 0$$

WANTQ = .FALSE. and WANTP = .FALSE.

$$lwork = \max(2 \times (\mathrm{M} - 1), 1) \quad \text{when NCOLB} = 0$$

$$lwork = \max(3 \times (\mathrm{M} - 1), 1) \quad \text{when NCOLB} > 0$$

On exit: WORK($\min(M, N)$) contains the total number of iterations taken by the QR algorithm.

The rest of the array is used as workspace.

16: IFAIL — INTEGER *Input/Output*

On entry: IFAIL must be set to 0, −1 or 1. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.

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

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 detected by the routine:

IFAIL = −1

One or more of the following conditions holds:

M < 0,
N < 0,
LDA < M,
NCOLB < 0,
LDB < M and NCOLB > 0,
LDQ < M and M < N and WANTQ = .TRUE.,
LDPT < N and M ≥ 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 SV(1), SV(2), ..., 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 $A = QEP^T$, where the leading $\min(m, n)$ by $\min(m, n)$ part of E is a bidiagonal matrix with SV(1), SV(2), ..., SV($\min(m, n)$) as the diagonal elements and WORK(1), WORK(2), ..., WORK($\min(m, n) - 1$) as the super-diagonal elements.

This failure is not likely to occur.

7 Accuracy

The computed factors Q, D and P satisfy the relation

$$QDP^T = A + E,$$

where

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

ϵ being 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 $\mathrm{SV}(k) < tol \times \mathrm{SV}(1)$, where l 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 Examples

For this routine two examples are presented, in Section 9.1 and Section 9.2. In the example programs distributed to sites, there is a single example program for F02WEF, with a main program:

```
*     F02WEF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*     .. External Subroutines ..
      EXTERNAL         EX1, EX2
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F02WEF Example Program Results'
      CALL EX1
      CALL EX2
      STOP
      END
```

The code to solve the two example problems is given in the subroutines EX1 and EX2, in Section 9.1 and Section 9.2 respectively.

9.1 Example 1

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

$$A = \begin{pmatrix} 2.0 & 2.5 & 2.5 \\ 2.0 & 2.5 & 2.5 \\ 1.6 & -0.4 & 2.8 \\ 2.0 & -0.5 & 0.5 \\ 1.2 & -0.3 & -2.9 \end{pmatrix}$$

together with the vector Q^Tb for the vector

$$b = \begin{pmatrix} 1.1 \\ 0.9 \\ 0.6 \\ 0.0 \\ -0.8 \end{pmatrix}$$

9.1.1 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.

```
*
      SUBROUTINE EX1
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          MMAX, NMAX, NCOLB
      PARAMETER        (MMAX=20,NMAX=10,NCOLB=1)
      INTEGER          LDA, LDB, LDPT
      PARAMETER        (LDA=MMAX,LDB=MMAX,LDPT=NMAX)
      INTEGER          LWORK
      PARAMETER        (LWORK=NMAX**2+5*(NMAX-1))
*     .. Local Scalars ..
      INTEGER          I, IFAIL, J, M, N
      LOGICAL          WANTP, WANTQ
*     .. Local Arrays ..
      real             A(LDA,NMAX), B(LDB), DUMMY(1), PT(LDPT,NMAX),
     +                 SV(NMAX), WORK(LWORK)
*     .. External Subroutines ..
      EXTERNAL         F02WEF
*     .. Executable Statements ..
      WRITE (NOUT,*)
      WRITE (NOUT,*)
      WRITE (NOUT,*) 'Example 1'
      WRITE (NOUT,*)
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*)
      READ (NIN,*)
      READ (NIN,*) M, N
      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 F02WEF(M,N,A,LDA,NCOLB,B,LDB,WANTQ,DUMMY,1,SV,WANTP,PT,
     +               LDPT,WORK,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,99998) (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,99998) (PT(J,I),J=1,N)
   40    CONTINUE
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Vector Q''*B'
         WRITE (NOUT,99998) (B(I),I=1,M)
      END IF
*
99999 FORMAT (1X,A,I5,A,I5)
99998 FORMAT (5(1X,F8.4))
      END
```

9.1.2 Program Data

```
F02WEF Example Program Data

Example 1
  5     3                       :Values of M and N

  2.0   2.5   2.5
  2.0   2.5   2.5
  1.6  -0.4   2.8
  2.0  -0.5   0.5
  1.2  -0.3  -2.9               :End of matrix A

  1.1   0.9   0.6   0.0  -0.8   :End of vector B
```

9.1.3 Program Results

```
 F02WEF Example Program Results


 Example 1

 Singular value decomposition of A

 Singular values
   6.5616   3.0000   2.4384

 Left-hand singular vectors, by column
   0.6011  -0.1961  -0.3165
   0.6011  -0.1961  -0.3165
   0.4166   0.1569   0.6941
   0.1688  -0.3922   0.5636
  -0.2742  -0.8629   0.0139

 Right-hand singular vectors, by column
   0.4694  -0.7845   0.4054
   0.4324  -0.1961  -0.8801
   0.7699   0.5883   0.2471

 Vector Q'*B
   1.6716   0.3922  -0.2276  -0.1000  -0.1000
```

9.2 Example 2

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

$$A = \begin{pmatrix} 2.0 & 2.0 & 1.6 & 2.0 & 1.2 \\ 2.5 & 2.5 & -0.4 & -0.5 & -0.3 \\ 2.5 & 2.5 & 2.8 & 0.5 & -2.9 \end{pmatrix}$$

9.2.1 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.

```
*
      SUBROUTINE EX2
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          MMAX, NMAX
      PARAMETER        (MMAX=10,NMAX=20)
      INTEGER          LDA, LDQ
      PARAMETER        (LDA=MMAX,LDQ=MMAX)
      INTEGER          LWORK
      PARAMETER        (LWORK=MMAX**2+5*(MMAX-1))
*     .. Local Scalars ..
      INTEGER          I, IFAIL, J, M, N, NCOLB
      LOGICAL          WANTP, WANTQ
*     .. Local Arrays ..
      real             A(LDA,NMAX), DUMMY(1), Q(LDQ,MMAX), SV(MMAX),
     +                 WORK(LWORK)
*     .. External Subroutines ..
      EXTERNAL         F02WEF
*     .. 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 F02WEF(M,N,A,LDA,NCOLB,DUMMY,1,WANTQ,Q,LDQ,SV,WANTP,DUMMY,
     +               1,WORK,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,99998) (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,99998) (A(J,I),J=1,M)
   40    CONTINUE
      END IF
*
99999 FORMAT (1X,A,I5,A,I5)
99998 FORMAT (5(1X,F8.4))
      END
```

9.2.2 Program Data

```
Example 2
  3     5                       :Values of M and N

  2.0   2.0   1.6   2.0   1.2
  2.5   2.5  -0.4  -0.5  -0.3
  2.5   2.5   2.8   0.5  -2.9   :End of matrix A
```

9.2.3 Program Results

```
 Example 2

 Singular value decomposition of A

 Singular values
   6.5616   3.0000   2.4384

 Left-hand singular vectors, by column
  -0.4694   0.7845  -0.4054
  -0.4324   0.1961   0.8801
  -0.7699  -0.5883  -0.2471

 Right-hand singular vectors, by column
  -0.6011   0.1961   0.3165
  -0.6011   0.1961   0.3165
  -0.4166  -0.1569  -0.6941
  -0.1688   0.3922  -0.5636
   0.2742   0.8629  -0.0139
```