NAG Fortran Library Routine Document

F02WDF

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

F02WDF returns the Householder QU factorization of a real rectangular m by $n(m \geq n)$ matrix A. Further, on request or if A is not of full rank, part or all of the singular value decomposition of A is returned.

2 Specification

```
      SUBROUTINE F02WDF(M, N, A, NRA, WANTB, B, TOL, SVD, IRANK, Z, SV, WANTR,
     1                  R, NRR, WANTPT, PT, NRPT, WORK, LWORK, IFAIL)
      INTEGER           M, N, NRA, IRANK, NRR, NRPT, LWORK, IFAIL
      real              A(NRA,N), B(M), TOL, Z(N), SV(N), R(NRR,N),
     1                  PT(NRPT,N), WORK(LWORK)
      LOGICAL           WANTB, SVD, WANTR, WANTPT, WANTPT
```

3 Description

The real m by $n(m \geq n)$ matrix A is first factorized as

$$A = Q\begin{pmatrix} U \\ 0 \end{pmatrix},$$

where Q is an m by m orthogonal matrix and U is an n by n upper triangular matrix.

If either U is singular or SVD is supplied as .TRUE., then the singular value decomposition (SVD) of U is obtained so that U is factorized as

$$U = RDP^T,$$

where R and P are n by n orthogonal matrices and D is the n by n diagonal matrix

$$D = \text{diag}(sv_1, sv_2, \ldots, sv_n),$$

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

Note that the SVD of A is then given by

$$A = Q_1\begin{pmatrix} D \\ 0 \end{pmatrix}P^T \quad \text{where} \quad Q_1 = Q\begin{pmatrix} R & 0 \\ 0 & I \end{pmatrix},$$

the diagonal elements of D being the singular values of A.

The option to form a vector Q^Tb, or if appropriate Q_1^Tb, is also provided.

The rank of the matrix A, based upon a user-supplied parameter TOL, is also returned.

The QU factorization of A is obtained by Householder transformations. To obtain the SVD of U the matrix is first reduced to bidiagonal form by means of plane rotations and then the QR algorithm is used to obtain the SVD of the bidiagonal form.

4 References

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: $M \geq N$.

2: N – INTEGER *Input*

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

Constraint: $1 \leq N \leq M$.

3: A(NRA,N) – ***real*** array *Input/Output*

On entry: the leading m by n part of A must contain the matrix to be factorized.

On exit: the leading m by n part of A, together with the n element vector Z, contains details of the Householder QU factorization.

Details of the storage of the QU factorization are given in Section 8.4.

4: NRA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F02WDF is called.

Constraint: $NRA \geq M$.

5: WANTB – LOGICAL *Input*

On entry: WANTB must be .TRUE. if $Q^T b$ or $Q_1^T b$ is required.

If on entry WANTB = .FALSE., then B is not referenced.

6: B(M) – ***real*** array *Input/Output*

On entry: if WANTB is supplied as .TRUE., then B must contain the m element vector b. Otherwise, B is not referenced.

On exit: B contains $Q_1^T b$ if SVD is returned as .TRUE. and $Q^T b$ if SVD is returned as .FALSE..

7: TOL – ***real*** *Input*

On entry: TOL must specify a relative tolerance to be used to determine the rank of A. TOL should be chosen as approximately the largest relative error in the elements of A. For example, if the elements of A are correct to about 4 significant figures, then TOL should be set to about 5×10^{-4}. See Section 8.3 for a description of how TOL is used to determine rank.

If TOL is outside the range $(\epsilon, 1.0)$, where ϵ is the ***machine precision***, then the value ϵ is used in place of TOL. For most problems this is unreasonably small.

8: SVD – LOGICAL *Input/Output*

On entry: SVD must be .TRUE. if the singular values are to be found even if A is of full rank.

If before entry, SVD = .FALSE. **and** A is determined to be of full rank, then only the QU factorization of A is computed.

On exit: SVD is returned as .FALSE. if only the QU factorization of A has been obtained and is returned as .TRUE. if the singular values of A have been obtained.

9: IRANK – INTEGER *Output*

On exit: IRANK returns the rank of the matrix A. (It should be noted that it is possible for IRANK to be returned as n and SVD to be returned as .TRUE., even if SVD was supplied as .FALSE.. This means that the matrix U only just failed the test for non-singularity.)

10: Z(N) – ***real*** array *Output*

On exit: the n element vector Z contains some details of the Householder transformations. See Section 8.4 for further information.

11: SV(N) – ***real*** array *Output*

On exit: if SVD is returned as .TRUE., SV contains the n singular values of A arranged in descending order.

12: WANTR – LOGICAL *Input*

On entry: WANTR must be .TRUE. if the orthogonal matrix R is required when the singular values are computed.

If on entry WANTR = .FALSE., then R is not referenced.

13: R(NRR,N) – ***real*** array *Output*

On exit: if SVD is returned as .TRUE. and WANTR was supplied as .TRUE., then the leading n by n part of R will contain the left-hand orthogonal matrix of the SVD of U.

14: NRR – INTEGER *Input*

On entry: the first dimension of the array R as declared in the (sub)program from which F02WDF is called.

Constraint: NRR $\geq$ N.

15: WANTPT – LOGICAL *Input*

On entry: WANTPT must be .TRUE. if the orthogonal matrix P^T is required when the singular values are computed.

Note that if SVD is returned as .TRUE., then PT is referenced even if WANTPT is supplied as .FALSE., but see parameter PT below.

16: PT(NRPT,N) – ***real*** array *Output*

On exit: if SVD is returned as .TRUE. and WANTPT was supplied as .TRUE., then the leading n by n part of PT contains the orthogonal matrix P^T. If SVD is returned as .TRUE., but WANTPT was supplied as .FALSE., then the leading n by n part of PT is used for internal workspace.

17: NRPT – INTEGER *Input*

On entry: the first dimension of the array PT as declared in the (sub)program from which F02WDF is called.

Constraint: NRPT $\geq$ N.

18: WORK(LWORK) – ***real*** array *Output*

If SVD is returned as .FALSE., then WORK(1) contains the condition number $\|U\|_E\|U^{-1}\|_E$ of the upper triangular matrix U.

If SVD is returned as .TRUE., then WORK(1) will contain the total number of iterations taken by the QR algorithm.

The rest of the array is used as workspace.

19: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F02WDF is called.

Constraint: LWORK $\geq 3 \times$ N.

20: 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 < 1,
or M $<$ N,
or NRA $<$ M,
or NRR $<$ N,
or NRPT $<$ N,
or LWORK $< 3 \times$ N.

(The routine only checks NRR if WANTR is supplied as .TRUE..)

IFAIL > 1

The QR algorithm has failed to converge to the singular values in $50 \times$ N iterations. In this case $\mathrm{SV}(1), \mathrm{SV}(2), \ldots, \mathrm{SV}(\mathrm{IFAIL}-1)$ may not have been correctly found and the remaining singular values may not be the smallest singular values. The matrix A has nevertheless been factorized as $A = Q_1 C P^T$, where C is an upper bidiagonal matrix with $\mathrm{SV}(1), \mathrm{SV}(2), \ldots, \mathrm{SV}(n)$ as its diagonal elements and $\mathrm{WORK}(2), \mathrm{WORK}(3), \ldots, \mathrm{WORK}(n)$ as its super-diagonal elements.

This failure cannot occur if SVD is returned as .FALSE. and in any case is extremely rare.

7 Accuracy

The computed factors Q, U, R, D and P^T satisfy the relations

$$Q\begin{pmatrix} U \\ 0 \end{pmatrix} = A + E,$$

$$Q\begin{pmatrix} R & 0 \\ 0 & I \end{pmatrix}\begin{pmatrix} D \\ 0 \end{pmatrix} P^T = A + F$$

where $\|E\|_2 \leq c_1 \epsilon \|A\|_2$, $\|F\|_2 \leq c_2 \epsilon \|A\|_2$,

ϵ being the ***machine precision*** and c_1 and c_2 are modest functions of m and n. Note that $\|A\|_2 = sv_1$.

8 Further Comments

8.1 Timing

The time taken by the routine to obtain the Householder QU factorization is approximately proportional to $n^2(3m - n)$.

The **additional** time taken to obtain the singular value decomposition is approximately proportional to n^3, where the constant of proportionality depends upon whether or not the orthogonal matrices R and P^T are required.

8.2 General Remarks

Singular vectors associated with a zero or multiple singular value, are not uniquely determined, even in exact arithmetic, and very different results may be obtained if they are computed on different machines.

This routine is column-biased and so is suitable for use in paged environments.

Unless otherwise stated in the Users' Note for your implementation, the routine may be called with the same actual array supplied for parameters A and PT, in which case, if SVD is returned as .TRUE., the leading n by n part of A is overwritten as specified for PT; also it may be called with the same array for parameters Z and SV, in which case, if SVD is returned as .TRUE., the singular values will overwrite the original contents of Z; also, if WANTPT = .FALSE., it may be called with the same array for parameters R and PT. However this is not standard Fortran 77, and may not work on all systems.

This routine is called by the least-squares routine F04JGF.

8.3 Determining the Rank of A

Following the QU factorization of A, if SVD is supplied as .FALSE., then the condition number of U given by

$$C(U) = \|U\|_F \|U^{-1}\|_F$$

is found, where $\|.\|_F$ denotes the Frobenius norm, and if $C(U)$ is such that

$$C(U) \times \text{TOL} > 1.0$$

then U is regarded as singular and the singular values of A are computed. If this test is not satisfied, then the rank of A is set to n. Note that if SVD is supplied as .TRUE. then this test is omitted.

When the singular values are computed, then the rank of A, r, is returned as the largest integer such that

$$sv_r > \text{TOL} \times sv_1,$$

unless $sv_1 = 0$ in which case r is returned as zero. That is, singular values which satisfy $sv_i \leq \text{TOL} \times sv_1$ are regarded as negligible because relative perturbations of order TOL can make such singular values zero.

8.4 Storage Details of the QU Factorization

The kth Householder transformation matrix, T_k, used in the QU factorization is chosen to introduce the zeros into the kth column and has the form

$$T_k = I - 2\begin{pmatrix} 0 \\ u \end{pmatrix}\begin{pmatrix} 0 & u^T \end{pmatrix}, \quad u^T u = 1,$$

where u is an $(m - k + 1)$ element vector.

In place of u the routine actually computes the vector z given by

$$z = 2u_1 u.$$

The first element of z is stored in Z(k) and the remaining elements of z are overwritten on the sub-diagonal elements of the kth column of A. The upper triangular matrix U is overwritten on the n by n upper triangular part of A.

9 Example

To obtain the rank and the singular value decomposition of the 6 by 4 matrix A given by

$$A = \begin{pmatrix} 22.25 & 31.75 & -38.25 & 65.50 \\ 20.00 & 26.75 & 28.50 & -26.50 \\ -15.25 & 24.25 & 27.75 & 18.50 \\ 27.25 & 10.00 & 3.00 & 2.00 \\ -17.25 & -30.75 & 11.25 & 7.50 \\ 17.25 & 30.75 & -11.25 & -7.50 \end{pmatrix}$$

the value TOL to be taken as 5×10^{-4}.

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.

```
*     F02WDF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          MMAX, NMAX, NRA, NRR, NRPT, LWORK
      PARAMETER        (MMAX=10,NMAX=8,NRA=MMAX,NRR=NMAX,NRPT=NMAX,
     +                 LWORK=3*NMAX)
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
      real             TOL
      INTEGER          I, IFAIL, IRANK, J, M, N
      LOGICAL          SVD
*     .. Local Arrays ..
      real             A(NRA,NMAX), PT(NRPT,NMAX), R(NRR,NMAX),
     +                 SV(NMAX), WORK(LWORK), Z(NMAX)
*     .. External Subroutines ..
      EXTERNAL         F02WDF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F02WDF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) M, N
      WRITE (NOUT,*)
      IF (N.LT.1 .OR. N.GT.NMAX .OR. M.LT.1 .OR. M.GT.MMAX) THEN
         WRITE (NOUT,99999) 'N or M out of range: N = ', N, '  M = ', M
         STOP
      END IF
      SVD = .TRUE.
      TOL = 5.0e-4
      READ (NIN,*) ((A(I,J),J=1,N),I=1,M)
      IFAIL = 0
*
      CALL F02WDF(M,N,A,NRA,.FALSE.,WORK,TOL,SVD,IRANK,Z,SV,.TRUE.,R,
     +            NRR,.TRUE.,PT,NRPT,WORK,LWORK,IFAIL)
*
      WRITE (NOUT,99999) 'Rank of A is', IRANK
      WRITE (NOUT,*)
      WRITE (NOUT,*) 'Details of QU factorization'
      DO 20 I = 1, M
         WRITE (NOUT,99998) (A(I,J),J=1,N)
   20 CONTINUE
      WRITE (NOUT,*)
      WRITE (NOUT,*) 'Vector Z'
      WRITE (NOUT,99998) (Z(I),I=1,N)
      WRITE (NOUT,*)
      WRITE (NOUT,*) 'Matrix R'
      DO 40 I = 1, N
         WRITE (NOUT,99998) (R(I,J),J=1,N)
   40 CONTINUE
      WRITE (NOUT,*)
```

```
      WRITE (NOUT,*) 'Singular values'
      WRITE (NOUT,99998) (SV(I),I=1,N)
      WRITE (NOUT,*)
      WRITE (NOUT,*) 'Matrix P**T'
      DO 60 I = 1, N
         WRITE (NOUT,99998) (PT(I,J),J=1,N)
   60 CONTINUE
      STOP
*
99999 FORMAT (1X,A,I5,A,I5)
99998 FORMAT (1X,8F9.3)
      END
```

9.2 Program Data

```
F02WDF Example Program Data
  6  4
  22.25  31.75 -38.25  65.50
  20.00  26.75  28.50 -26.50
 -15.25  24.25  27.75  18.50
  27.25  10.00   3.00   2.00
 -17.25 -30.75  11.25   7.50
  17.25  30.75 -11.25  -7.50
```

9.3 Program Results

```
 F02WDF Example Program Results

 Rank of A is    4

 Details of QU factorization
   -49.652  -44.409   20.354   -8.882
     0.403  -48.277   -9.589  -20.376
    -0.307    0.837   52.927  -48.881
     0.549   -0.391   -0.836  -50.674
    -0.347   -0.258   -0.185    0.632
     0.347    0.258    0.185   -0.632

 Vector Z
     1.448    1.115    1.482    1.448

 Matrix R
    -0.564    0.634    0.423    0.317
    -0.351    0.395   -0.679   -0.509
    -0.640   -0.569    0.309   -0.413
    -0.386   -0.343   -0.514    0.685

 Singular values
    91.000   68.250   45.500   22.750

 Matrix P**T
     0.308    0.462   -0.462    0.692
    -0.462   -0.692   -0.308    0.462
    -0.462    0.308    0.692    0.462
    -0.692    0.462   -0.462   -0.308
```