NAG 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, LDA, WANTB, B, TOL, SVD, IRANK, Z, SV,          &
                   WANTR, R, LDR, WANTPT, PT, LDPT, WORK, LWORK, IFAIL)

INTEGER            M, N, LDA, IRANK, LDR, LDPT, LWORK, IFAIL
REAL (KIND=nag_wp) A(LDA,N), B(M), TOL, Z(N), SV(N), R(LDR,N),              &
                   PT(LDPT,N), WORK(LWORK)
LOGICAL            WANTB, SVD, WANTR, 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

$$U = RDP^{\mathrm{T}},$$

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

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

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

Note that the SVD of A is then given by

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

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

The option to form a vector $Q^{\mathrm{T}}b$, or if appropriate $Q_1^{\mathrm{T}}b$, is also provided.

The rank of the matrix A, based upon a user-supplied argument 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 Arguments

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(LDA, N) – REAL (KIND=nag_wp) 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 9.4.

4: LDA – INTEGER *Input*

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

Constraint: LDA $\geq$ M.

5: WANTB – LOGICAL *Input*

On entry: must be .TRUE. if $Q^{\mathrm{T}}b$ or $Q_1^{\mathrm{T}}b$ is required.

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

6: B(M) – REAL (KIND=nag_wp) array *Input/Output*

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

On exit: contains $Q_1^{\mathrm{T}}b$ if SVD is returned as .TRUE. and $Q^{\mathrm{T}}b$ if SVD is returned as .FALSE..

7: TOL – REAL (KIND=nag_wp) *Input*

On entry: 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, TOL should be set to about 5×10^{-4}. See Section 9.3 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***, the value ϵ is used in place of TOL. For most problems this is unreasonably small.

8: SVD – LOGICAL *Input/Output*

On entry: 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, only the QU factorization of A is computed.

On exit: 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: 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 nonsingularity.)

10: Z(N) – REAL (KIND=nag_wp) array *Output*

On exit: the n-element vector Z contains some details of the Householder transformations. See Section 9.4 for further information.

11: SV(N) – REAL (KIND=nag_wp) 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: must be .TRUE. if the orthogonal matrix R is required when the singular values are computed.

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

13: R(LDR, N) – REAL (KIND=nag_wp) array *Output*

Note: the second dimension of the array R must be at least N if WANTR = .TRUE., and at least 1 otherwise.

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

14: LDR – INTEGER *Input*

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

Constraints:

if WANTR = .TRUE., LDR $\geq$ N;
otherwise LDR $\geq$ 1.

15: WANTPT – LOGICAL *Input*

On entry: 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., PT is referenced even if WANTPT is supplied as .FALSE., but see argument PT.

16: PT(LDPT, N) – REAL (KIND=nag_wp) array *Output*

On exit: if SVD is returned as .TRUE. and WANTPT was supplied as .TRUE., 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., the leading n by n part of PT is used for internal workspace.

17: LDPT – INTEGER *Input*

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

Constraint: LDPT $\geq$ N.

18: WORK(LWORK) – REAL (KIND=nag_wp) array *Output*

On exit: if SVD is returned as .FALSE., WORK(1) contains the condition number $\|U\|_E \left\|U^{-1}\right\|_E$ of the upper triangular matrix U.

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

The rest of the array is used as workspace and so contains no meaningful information.

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. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation for details.

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 argument, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL $= 0$ unless the routine detects an error or a warning has been flagged (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 or warnings detected by the routine:

IFAIL $= 1$

On entry, N $<$ 1,
or M $<$ N,
or LDA $<$ M,
or LDR $<$ N when WANTR $=$ .TRUE.,
or LDPT $<$ N
or LWORK $< 3 \times$ N.

(The routine only checks LDR 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 $A = Q_1 C P^{\mathrm{T}}$, 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 superdiagonal elements.

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

IFAIL $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

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

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

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

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 Parallelism and Performance

F02WDF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F02WDF makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

9.1 Timing

The time taken by F02WDF 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.

9.2 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.

Unless otherwise stated in the Users' Note for your implementation, the routine may be called with the same array for arguments 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 arguments R and PT. However this is not standard Fortran, and may not work on all systems.

This routine is called by the least squares routine F04JGF.

9.3 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

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

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.

9.4 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

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

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 subdiagonal elements of the kth column of A. The upper triangular matrix U is overwritten on the n by n upper triangular part of A.

10 Example

This example obtains 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}.

10.1 Program Text

```
    Program f02wdfe

!     F02WDF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: f02wdf, nag_wp, x04cbf
!     .. Implicit None Statement ..
      Implicit None
```

```
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: tol
      Integer                          :: i, ifail, irank, lda, ldpt, ldr,     &
                                          lwork, m, n
      Logical                          :: svd, wantb, wantpt, wantr
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:), pt(:,:), r(:,:), sv(:),      &
                                          work(:), z(:)
      Character (1)                    :: clabs(1), rlabs(1)
!     .. Executable Statements ..
      Write (nout,*) 'F02WDF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) m, n
      Write (nout,*)
      lda = m
      ldpt = n
      ldr = n
      lwork = 3*n
      Allocate (a(lda,n),pt(ldpt,n),r(ldr,n),sv(n),work(lwork),z(n))
      svd = .True.
      tol = 5.0E-4_nag_wp
      Read (nin,*)(a(i,1:n),i=1,m)
      wantb = .False.
      wantr = .True.
      wantpt = .True.

!     ifail: behaviour on error exit
!             =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
      ifail = 0
      Call f02wdf(m,n,a,lda,wantb,work,tol,svd,irank,z,sv,wantr,r,ldr,wantpt,  &
        pt,ldpt,work,lwork,ifail)

      Write (nout,99999) 'Rank of A is', irank
      Write (nout,*)
      Flush (nout)

      ifail = 0
      Call x04cbf('General',' ',m,n,a,lda,'F9.3','Details of QU factorization' &
        ,'N',rlabs,'N',clabs,80,0,ifail)

      Write (nout,*)
      Write (nout,*) 'Vector Z'
      Write (nout,99998) z(1:n)
      Write (nout,*)
      Flush (nout)

      ifail = 0
      Call x04cbf('General',' ',n,n,r,ldr,'F9.3','Matrix R','N',rlabs,'N',     &
        clabs,80,0,ifail)

      Write (nout,*)
      Write (nout,*) 'Singular values'
      Write (nout,99998) sv(1:n)
      Write (nout,*)
      Flush (nout)

      ifail = 0
      Call x04cbf('General',' ',n,n,pt,ldpt,'F9.3','Matrix P**T','N',rlabs,    &
        'N',clabs,80,0,ifail)

99999 Format (1X,A,I5,A,I5)
99998 Format (1X,8F9.3)
    End Program f02wdfe
```

10.2 Program Data

```
F02WDF Example Program Data
  6  4                              : m, n
  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    : matrix A
```

10.3 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
```