NAG Library Routine Document

F02WUF

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

F02WUF returns all, or part, of the singular value decomposition of a real upper triangular matrix.

2 Specification

```
SUBROUTINE F02WUF (N, A, LDA, NCOLB, B, LDB, WANTQ, Q, LDQ, SV, WANTP,     &
                   WORK, IFAIL)
INTEGER            N, LDA, NCOLB, LDB, LDQ, IFAIL
REAL (KIND=nag_wp) A(LDA,*), B(LDB,*), Q(LDQ,*), SV(N), WORK(*)
LOGICAL            WANTQ, WANTP
```

3 Description

The n by n upper triangular matrix R is factorized as

$$R = QSP^{\mathrm{T}},$$

where Q and P are n by n orthogonal matrices and S is an n by n diagonal matrix with non-negative diagonal elements, $\sigma_1, \sigma_2, \ldots, \sigma_n$, ordered such that

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0.$$

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^{T} may be requested and the matrix C given by

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

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 Chan (1982), Dongarra *et al.* (1979), Golub and Van Loan (1996), Hammarling (1985) and Wilkinson (1978).

Note that if K is any orthogonal diagonal matrix so that

$$KK^{\mathrm{T}} = I$$

(that is the diagonal elements of K are $+1$ or -1) then

$$A = (QK)S(PK)^{\mathrm{T}}$$

is also a singular value decomposition of A.

4 References

Chan T F (1982) An improved algorithm for computing the singular value decomposition *ACM Trans. Math. Software* **8** 72–83

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

Golub G H and Van Loan C F (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

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: n, the order of the matrix R.

If $\mathrm{N} = 0$, an immediate return is effected.

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

2: $\mathrm{A}(\mathrm{LDA}, *)$ – REAL (KIND=nag_wp) array *Input/Output*

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

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., the n by n part of A will contain the n by n orthogonal matrix P^{T}, 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 F02WUF is called.

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

4: NCOLB – INTEGER *Input*

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

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

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

5: $\mathrm{B}(\mathrm{LDB}, *)$ – REAL (KIND=nag_wp) array *Input/Output*

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

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

On exit: the leading n by $ncolb$ part of the array B is overwritten by the matrix $Q^{\mathrm{T}}B$.

6: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F02WUF is called.

Constraints:

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

7: WANTQ – LOGICAL *Input*

On entry: must be .TRUE. if the matrix Q is required.

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

8: Q(LDQ, $*$) – REAL (KIND=nag_wp) array *Output*

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

On exit: with WANTQ = .TRUE., the leading n by n part of the array Q will contain the orthogonal 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 F02WUF is called.

Constraints:

if WANTQ = .TRUE., LDQ $\geq \max(1, \mathrm{N})$;
otherwise LDQ ≥ 1.

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

On exit: the array SV will contain the n diagonal elements of the matrix S.

11: WANTP – LOGICAL *Input*

On entry: must be .TRUE. if the matrix P^{T} is required, in which case P^{T} is overwritten on the array A, otherwise WANTP must be .FALSE..

12: WORK($*$) – REAL (KIND=nag_wp) array *Output*

Note: the dimension of the array WORK must be at least $\max(1, 2 \times (\mathrm{N}-1))$ if NCOLB $= 0$ and WANTQ = .FALSE. and WANTP = .FALSE., $\max(1, 3 \times (\mathrm{N}-1))$ if (NCOLB $= 0$ and WANTQ = .FALSE. and WANTP = .TRUE.) or (WANTP = .FALSE. and (NCOLB > 0 or WANTQ = .TRUE.)), and at least $\max(1, 5 \times (\mathrm{N}-1))$ otherwise.

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

The rest of the array is used as internal workspace.

13: 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 Section 3.3 in the Essential Introduction 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 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.**

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 < 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 $50 \times \mathrm{N}$ 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 $R = QEP^{\mathrm{T}}$, where E is a bidiagonal matrix with $\mathrm{SV}(1), \mathrm{SV}(2), \ldots, \mathrm{SV}(n)$ as the diagonal elements and $\mathrm{WORK}(1), \mathrm{WORK}(2), \ldots, \mathrm{WORK}(n-1)$ as the superdiagonal elements.

This failure is not likely to occur.

IFAIL $= -99$

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

See Section 3.8 in the Essential Introduction for further information.

IFAIL $= -399$

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

See Section 3.7 in the Essential Introduction for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

7 Accuracy

The computed factors Q, S and P satisfy the relation

$$QSP^{\mathrm{T}} = R + E,$$

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 $\|A\| = \mathrm{SV}(1)$.

A similar result holds for the computed matrix $Q^{\mathrm{T}}B$.

The computed matrix Q satisfies the relation

$$Q = T + F,$$

where T is exactly orthogonal and

$$\|F\| \leq d\epsilon,$$

where d is a modest function of n. A similar result holds for P.

8 Parallelism and Performance

F02WUF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F02WUF 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

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 $\mathrm{SV}(k) < tol \times \mathrm{SV}(1)$, and 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 .

10 Example

This example finds the singular value decomposition of the 3 by 3 upper triangular matrix

$$A = \begin{pmatrix} -4 & -2 & -3 \\ 0 & -3 & -2 \\ 0 & 0 & -4 \end{pmatrix},$$

together with the vector $Q^{\mathrm{T}}b$ for the vector

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

10.1 Program Text

```
    Program f02wufe

!     F02WUF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: f02wuf, nag_wp, x04cbf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, lda, ldb, ldq, lwork, n,   &
                                          ncolb
      Logical                          :: wantp, wantq
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:), b(:), q(:,:), sv(:), work(:)
      Character (1)                    :: clabs(1), rlabs(1)
!     .. Executable Statements ..
      Write (nout,*) 'F02WUF Example Program Results'
      Write (nout,*)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n, ncolb
      lda = n
      ldb = n
      ldq = n
      lwork = 5*(n-1)
      Allocate (a(lda,n),b(ldb),q(ldq,n),sv(n),work(lwork))
      Read (nin,*)(a(i,i:n),i=1,n)
      Read (nin,*) b(1:n)
```

```
      wantq = .True.
      wantp = .True.

!     ifail: behaviour on error exit
!             =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
      ifail = 0
!     Find the SVD of A
      Call f02wuf(n,a,lda,ncolb,b,ldb,wantq,q,ldq,sv,wantp,work,ifail)

      Write (nout,*) 'Singular value decomposition of A'
      Write (nout,*)
      Write (nout,*) 'Singular values'
      Write (nout,99999) sv(1:n)
      Write (nout,*)
      Flush (nout)

      ifail = 0
      Call x04cbf('General',' ',n,n,q,ldq,'F8.4', &
        'Left-hand singular vectors, by column','N',rlabs,'N',clabs,80,0, &
        ifail)

      Write (nout,*)
      Write (nout,*) 'Right-hand singular vectors, by column'
      Do i = 1, n
        Write (nout,99999) a(1:n,i)
      End Do
      Write (nout,*)
      Write (nout,*) 'Vector Q''*B'
      Write (nout,99999) b(1:n)

99999 Format (1X,3(1X,F8.4))
    End Program f02wufe
```

10.2 Program Data

```
F02WUF Example Program Data
   3, 1                : n, ncolb
  -4.0  -2.0  -3.0
        -3.0  -2.0
              -4.0   : matrix A
  -1.0  -1.0  -1.0   : vector B
```

10.3 Program Results

```
 F02WUF Example Program Results

 Singular value decomposition of A

 Singular values
    6.5616   3.0000   2.4384

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

 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
```