

NAG Library Routine Document

F01BLF

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

F01BLF calculates the rank and pseudo-inverse of an m by n real matrix, $m \geq n$, using a QR factorization with column interchanges.

2 Specification

SUBROUTINE F01BLF (M, N, T, A, LDA, AIJMAX, IRANK, INC, D, U, LDU, DU, &
IFAIL)

INTEGER M, N, LDA, IRANK, INC(N), LDU, IFAIL
REAL (KIND=nag_wp) T, A(LDA,N), AIJMAX(N), D(M), U(LDU,N), DU(N)

3 Description

Householder's factorization with column interchanges is used in the decomposition $F = QU$, where F is A with its columns permuted, Q is the first r columns of an m by m orthogonal matrix and U is an r by n upper-trapezoidal matrix of rank r . The pseudo-inverse of F is given by X where

$$X = U^T(UU^T)^{-1}Q^T.$$

If the matrix is found to be of maximum rank, $r = n$, U is a nonsingular n by n upper-triangular matrix and the pseudo-inverse of F simplifies to $X = U^{-1}Q^T$. The transpose of the pseudo-inverse of A is overwritten on A .

4 References

Peters G and Wilkinson J H (1970) The least squares problem and pseudo-inverses *Comput. J.* **13** 309–316

Wilkinson J H and Reinsch C (1971) *Handbook for Automatic Computation II, Linear Algebra* Springer-Verlag

5 Parameters

1: M – INTEGER *Input*
2: N – INTEGER *Input*

On entry: m and n , the number of rows and columns in the matrix A .

Constraint: $M \geq N$.

3: T – REAL (KIND=nag_wp) *Input*

On entry: the tolerance used to decide when elements can be regarded as zero (see Section 9).

4: A(LDA,N) – REAL (KIND=nag_wp) array *Input/Output*

On entry: the m by n rectangular matrix A .

On exit: the transpose of the pseudo-inverse of A .

- 5: LDA – INTEGER *Input*
On entry: the first dimension of the array A as declared in the (sub)program from which F01BLF is called.
Constraint: $LDA \geq M$.
- 6: AIJMAX(N) – REAL (KIND=nag_wp) array *Output*
On exit: AIJMAX(*i*) contains the element of largest modulus in the reduced matrix at the *i*th stage. If $r < n$, then only the first $r + 1$ elements of AIJMAX have values assigned to them; the remaining elements are unused. The ratio AIJMAX(1)/AIJMAX(*r*) usually gives an indication of the condition number of the original matrix (see Section 9).
- 7: IRANK – INTEGER *Output*
On exit: *r*, the rank of A as determined using the tolerance T.
- 8: INC(N) – INTEGER array *Output*
On exit: the record of the column interchanges in the Householder factorization.
- 9: D(M) – REAL (KIND=nag_wp) array *Workspace*
 10: U(LDU, N) – REAL (KIND=nag_wp) array *Workspace*
- 11: LDU – INTEGER *Input*
On entry: the first dimension of the array U as declared in the (sub)program from which F01BLF is called.
Constraint: $LDU \geq N$.
- 12: DU(N) – REAL (KIND=nag_wp) array *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

Inverse not found, due to an incorrect determination of IRANK (see Section 9).

IFAIL = 2

Invalid tolerance, due to

- (i) T is negative, IRANK = -1;
- (ii) T too large, IRANK = 0;
- (iii) T too small, IRANK > 0.

IFAIL = 3

On entry, $M < N$.

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

For most matrices the pseudo-inverse is the best possible having regard to the condition of A and the choice of T . Note that only the singular value decomposition method can be relied upon to give maximum accuracy for the precision of computation used and correct determination of the condition of a matrix (see Wilkinson and Reinsch (1971)).

The computed factors Q and U satisfy the relation $QU = F + E$ where

$$\|E\|_2 < c\epsilon\|A\|_2 + \eta\sqrt{(m-r)(n-r)}$$

in which c is a modest function of m and n , η is the value of T , and ϵ is the *machine precision*.

8 Parallelism and Performance

Not applicable.

9 Further Comments

The time taken by F01BLF is approximately proportional to mnr .

The most difficult practical problem is the determination of the rank of the matrix (see pages 314–315 of Peters and Wilkinson (1970)); only the singular value decomposition method gives a reliable indication of rank deficiency (see pages 134–151 of Wilkinson and Reinsch (1971) and F08KBF (DGESVD)). In F01BLF a tolerance, T , is used to recognize ‘zero’ elements in the remaining matrix at each step in the factorization. The value of T should be set at n times the bound on possible errors in individual elements of the original matrix. If the elements of A vary widely in their orders of magnitude, of course this presents severe difficulties. Sound decisions can only be made by somebody who appreciates the underlying physical problem.

If the condition number of A is 10^p we expect to get p figures wrong in the pseudo-inverse. An estimate of the condition number is usually given by $\text{AIJMAX}(1)/\text{AIJMAX}(r)$.

10 Example

A complete program follows which outputs the maximum of the moduli of the ‘remaining’ elements at each step in the factorization, the rank, as determined by the given value of T, and the transposed pseudo-inverse. Data and results are given for an example which is a 6 by 5 matrix of deficient rank in which the last column is a linear combination of the other four. Setting T to ϵ times the norm of the matrix, the rank is correctly determined as 4 and the pseudo-inverse is computed to full implementation accuracy.

10.1 Program Text

```

Program f01blfe

!      F01BLF Example Program Text

!      Mark 25 Release. NAG Copyright 2014.

!      .. Use Statements ..
Use nag_library, Only: f01blf, f06raf, nag_wp, x02ajf, x04cbf
!      .. Implicit None Statement ..
Implicit None
!      .. Parameters ..
Integer, Parameter          :: indent = 0, ncols = 80, nin = 5,      &
                             nout = 6
Character (1), Parameter   :: diag = 'N', matrix = 'G', nolabel = &
                             'N'
Character (8), Parameter   :: form = '1P,E12.4'
!      .. Local Scalars ..
Real (Kind=nag_wp)        :: anorm, t
Integer                   :: i, ifail, irank, lda, ldu, m, n
Character (9)              :: norm
Character (27)             :: title
!      .. Local Arrays ..
Real (Kind=nag_wp), Allocatable :: a(:,,:), aijmax(:), d(:), du(:), u(:,,:)
Real (Kind=nag_wp)          :: work(1)
Integer, Allocatable        :: inc(:)
Character (1)               :: dummy(1)
!      .. Intrinsic Procedures ..
Intrinsic                   :: min
!      .. Executable Statements ..
Write (nout,*) 'F01BLF Example Program Results'
Write (nout,*)
!      Skip heading in data file
Read (nin,*)
Read (nin,*) m, n
lda = m
ldu = n
Allocate (a(lda,n),aijmax(n),d(m),du(n),u(ldu,n),inc(n))
Read (nin,*)(a(i,1:n),i=1,m)
!      Set t = eps times norm of A.
norm = 'Frobenius'
anorm = f06raf(norm,m,n,a,lda,work)
t = anorm*x02ajf()

!      ifail: behaviour on error exit
!              =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
ifail = 0
Call f01blf(m,n,t,a,lda,aijmax,irank,inc,d,u,ldu,du,ifail)

Write (nout,*) 'Maximum element in A(K) for I.GE.K and J.GE.K'
Write (nout,*)
Write (nout,*) '      K      Modulus'
Write (nout,99999)(i,aijmax(i),i=1,min(n,irank+1))
Write (nout,*)
Write (nout,99998) 'Rank = ', irank
Write (nout,*)
Write (nout,99997) 'T = ', t, ' (machine dependent)'
Write (nout,*)
Flush (nout)

```

```

!      Print the result matrix A.
      title = 'Transpose of pseudo-inverse'
      ifail = 0
      Call x04cbf(matrix,diag,m,n,a,lda,form,title,nolabel,dummy,nolabel, &
        dummy,ncols,indent,ifail)

99999 Format (1X,I4,2X,1P,E12.4)
99998 Format (1X,A,I5)
99997 Format (1X,A,1P,E11.4,A)
      End Program f01blfe

```

10.2 Program Data

```

F01BLF Example Program Data
  6  5                               : m, n
  7.0 -2.0  4.0  9.0  1.8
  3.0  8.0 -4.0  6.0  1.3
  9.0  6.0  1.0  5.0  2.1
 -8.0  7.0  5.0  2.0  0.6
  4.0 -1.0  2.0  8.0  1.3
  1.0  6.0  3.0 -5.0  0.5         : a

```

10.3 Program Results

F01BLF Example Program Results

Maximum element in A(K) for I.GE.K and J.GE.K

K	Modulus
1	9.0000E+00
2	9.3101E+00
3	8.7461E+00
4	5.6832E+00
5	2.8449E-16

Rank = 4

T = 2.9948E-15 (machine dependent)

Transpose of pseudo-inverse

1.7807E-02	-2.1565E-02	5.2029E-02	2.3686E-02	7.1957E-03
-1.1826E-02	4.3417E-02	-8.1265E-02	3.5717E-02	-1.3957E-03
4.7157E-02	2.9446E-02	1.3926E-02	-1.3808E-02	7.6720E-03
-5.6636E-02	2.9132E-02	4.7442E-02	3.0478E-02	5.0415E-03
-3.6741E-03	-1.3781E-02	1.6647E-02	3.5665E-02	3.4857E-03
3.8408E-02	3.4256E-02	5.7594E-02	-5.7134E-02	7.3123E-03
