

NAG Fortran Library Routine Document

F04LHF

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

F04LHF calculates the approximate solution of a set of real linear equations with multiple right-hand sides, $AX = B$ or $A^T X = B$, where A is an almost block-diagonal matrix which has been factorized by F01LHF.

2 Specification

```

SUBROUTINE F04LHF(TRANS, N, NBLOKS, BLKSTR, A, LENA, PIVOT, B, LDB, IR,
1                IFAIL)
  INTEGER          N, NBLOKS, BLKSTR(3,NBLOKS), LENA, PIVOT(N), LDB, IR,
1                IFAIL
  real            A(LENA), B(LDB,IR)
  CHARACTER*1     TRANS

```

3 Description

The routine solves a set of real linear equations $AX = B$ or $A^T X = B$, where A is almost block-diagonal. A must first be factorized by F01LHF. F04LHF then computes X by forward and backward substitution over the blocks.

4 References

Diaz J C, Fairweather G and Keast P (1983) Fortran packages for solving certain almost block diagonal linear systems by modified alternate row and column elimination *ACM Trans. Math. Software* **9** 358–375

5 Parameters

- 1: TRANS – CHARACTER*1 *Input*
On entry: TRANS specifies the equations to be solved as follows:
 if TRANS = 'N', solve $AX = B$;
 if TRANS = 'T', solve $A^T X = B$.
- 2: N – INTEGER *Input*
On entry: n , the order of the matrix A .
Constraint: $N > 0$.
- 3: NBLOKS – INTEGER *Input*
On entry: the total number of blocks of the matrix A , as supplied to F01LHF.
Constraint: $0 < \text{NBLOKS} \leq N$.
- 4: BLKSTR(3,NBLOKS) – INTEGER array *Input*
On entry: information which describes the block structure of A , as supplied to F01LHF.

- 5: A(LENA) – *real* array *Input*
On entry: the elements in the factorization of A , as returned by F01LHF.
- 6: LENA – INTEGER *Input*
On entry: the dimension of the array A as declared in the (sub)program from which F04LHF is called.
Constraint: $LENA \geq \sum_{k=1} NBLOKSBLKSTR(1, k) \times BLKSTR(2, k)$.
- 7: PIVOT(N) – INTEGER array *Input*
On entry: details of the interchanges in the factorization, as returned by F01LHF.
- 8: B(LDB,IR) – *real* array *Input/Output*
On entry: the n by r right-hand side matrix B .
On exit: B is overwritten by the n by r solution matrix X .
- 9: LDB – INTEGER *Input*
On entry: the first dimension of the array B as declared in the (sub)program from which F04LHF is called.
Constraint: $LDB \geq N$.
- 10: IR – INTEGER *Input*
On entry: r , the number of right-hand sides.
Constraint: $IR > 0$.
- 11: 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 $NBLOKS < 1$,
- or $IR < 1$,
- or $LDB < N$,
- or $N < NBLOKS$,
- or LENA is too small,
- or illegal values detected in BLKSTR,
- or $TRANS \neq 'N'$ or $'T'$.


```

*   .. Local Scalars ..
real          TOL
INTEGER        I, IFAIL, INDEX, IR, J, K, N, NBASEK, NBLOKS
*   .. Local Arrays ..
real          A(LENA), B(LDB,IRMAX)
INTEGER        BLKSTR(3,NBLMAX), PIVOT(NMAX)
*   .. External Subroutines ..
EXTERNAL       F01LHF, F04LHF
*   .. Executable Statements ..
WRITE (NOUT,*) 'F04LHF Example Program Results'
*   Skip heading in data file
READ (NIN,*)
READ (NIN,*) NBLOKS
WRITE (NOUT,*)
IF (NBLOKS.LE.NBLMAX) THEN
  NBASEK = 0
  N = 0
  DO 40 I = 1, NBLOKS
    READ (NIN,*) (BLKSTR(J,I),J=1,3)
    DO 20 K = 1, BLKSTR(1,I)
      IF (NBASEK+BLKSTR(2,I)*BLKSTR(1,I).GT.LENA) THEN
+       WRITE (NOUT,*)
+         ' Array A is too small for this problem'
        STOP
      ELSE
+       READ (NIN,*) (A(NBASEK+(J-1)*BLKSTR(1,I)+K),J=1,
+         BLKSTR(2,I))
        END IF
20    CONTINUE
    NBASEK = NBASEK + BLKSTR(2,I)*BLKSTR(1,I)
    N = N + BLKSTR(1,I)
40    CONTINUE
    IF (N.GT.NMAX) THEN
      WRITE (NOUT,*) ' N is too large'
      STOP
    END IF
    TOL = 0.0e0
    IFAIL = -1
*
*   CALL F01LHF(N,NBLOKS,BLKSTR,A,LENA,PIVOT,TOL,INDEX,IFAIL)
*
    IF (IFAIL.EQ.0) THEN
      READ (NIN,*) IR
      IF (IR.LE.IRMAX) THEN
        READ (NIN,*) ((B(I,J),I=1,N),J=1,IR)
        IFAIL = -1
*
*   CALL F04LHF('N',N,NBLOKS,BLKSTR,A,LENA,PIVOT,B,LDB,IR,
+   IFAIL)
*
      IF (IFAIL.EQ.0) THEN
        WRITE (NOUT,*) 'Component Solution'
        DO 60 I = 1, N
          WRITE (NOUT,99999) I, (B(I,J),J=1,IR)
60        CONTINUE
        END IF
      ELSE
        WRITE (NOUT,*) ' Too many right hand sides speciûed'
      END IF
    END IF
  ELSE
    WRITE (NOUT,*) ' NBLOKS is invalid'
  END IF
  STOP
*
99999 FORMAT (1X,I5,6X,5F6.4)
END

```

9.2 Program Data

```

F04LHF Example Program Data
  5          : Number of blocks
  2 4 3     : Number of rows, columns and column overlap, block 1
-1.00 -0.98 -0.79 -0.15
-1.00 0.25 -0.87 0.35          : End block 1
  4 7 4     : Number of rows, columns and column overlap, block 2
  0.78 0.31 -0.85 0.89 -0.69 -0.98 -0.76
-0.82 0.12 -0.01 0.75 0.32 -1.00 -0.53
-0.83 -0.98 -0.58 0.04 0.87 0.38 -1.00
-0.21 -0.93 -0.84 0.37 -0.94 -0.96 -1.00          : End block 2
  5 8 2     : Number of rows, columns and column overlap, block 3
-0.99 -0.91 -0.28 0.90 0.78 -0.93 -0.76 0.48
-0.87 -0.14 -1.00 -0.59 -0.99 0.21 -0.73 -0.48
-0.93 -0.91 0.10 -0.89 -0.68 -0.09 -0.58 -0.21
  0.85 -0.39 0.79 -0.71 0.39 -0.99 -0.12 -0.75
  0.17 -1.37 1.29 -1.59 1.10 -1.63 -1.01 -0.27          : End block 3
  3 6 3     : Number of rows, columns and column overlap, block 4
  0.08 0.61 0.54 -0.41 0.16 -0.46
-0.67 0.56 -0.99 0.16 -0.16 0.98
-0.24 -0.41 0.40 -0.93 0.70 0.43          : End block 4
  4 5 0     : Number of rows, columns and column overlap, block 5
  0.71 -0.97 -0.60 -0.30 0.18
-0.47 -0.98 -0.73 0.07 0.04
-0.25 -0.92 -0.52 -0.46 -0.58
  0.89 -0.94 -0.54 -1.00 -0.36          : End block 5
  1          : Number of right hand sides
-2.92 -1.27 -1.30 -1.17 -2.10 -4.51 -1.71 -4.59
-4.19 -0.93 -3.31 0.52 -0.12 -0.05 -0.98 -2.07
-2.73 -1.95          : End right hand side 1

```

9.3 Program Results

F04LHF Example Program Results

```

Component Solution
  1          1.0000
  2          1.0000
  3          1.0000
  4          1.0000
  5          1.0000
  6          1.0000
  7          1.0000
  8          1.0000
  9          1.0000
 10          1.0000
 11          1.0000
 12          1.0000
 13          1.0000
 14          1.0000
 15          1.0000
 16          1.0000
 17          1.0000
 18          1.0000

```
