NAG Toolbox for MATLAB

f04lh

1 Purpose

f04lh calculates the approximate solution of a set of real linear equations with multiple right-hand sides, $AX = B$ or $A^{\mathrm{T}}X = B$, where A is an almost block-diagonal matrix which has been factorized by f01lh.

2 Syntax

```
[b, ifail] = f04lh(trans, blkstr, a, pivot, b, 'n', n, 'nbloks', nbloks, 'lena', lena, 'ir', ir)
```

3 Description

f04lh solves a set of real linear equations $AX = B$ or $A^{\mathrm{T}}X = B$, where A is almost block-diagonal. A must first be factorized by f01lh. f04lh 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

5.1 Compulsory Input Parameters

1: **trans – string**

Specifies the equations to be solved.

trans = 'N'

Solve $AX = B$.

trans = 'T'

Solve $A^{\mathrm{T}}X = B$.

Constraint: **trans** = 'N' or 'T'.

2: **blkstr**(**3**,**nbloks**) **– int32 array**

Information which describes the block structure of A, as supplied to f01lh.

3: **a**(**lena**) **– double array**

The elements in the factorization of A, as returned by f01lh.

4: **pivot**(**n**) **– int32 array**

Details of the interchanges in the factorization, as returned by f01lh.

5: **b**(**ldb**,**ir**) **– double array**

ldb, the first dimension of the array, must be at least **n**.

The n by r right-hand side matrix B.

5.2 Optional Input Parameters

1: **n – int32 scalar**

Default: The dimension of the array **pivot**.

n, the order of the matrix A.

Constraint: $\mathbf{n} > 0$.

2: **nbloks – int32 scalar**

Default: The dimension of the array **blkstr**.

the total number of blocks of the matrix A, as supplied to f01lh.

Constraint: $0 < \mathbf{nbloks} \leq \mathbf{n}$.

3: **lena – int32 scalar**

Default: The dimension of the array **a**.

Constraint: $\mathbf{lena} \geq \sum_{k=1}^{\mathbf{nbloks}} \mathbf{blkstr}(1,k) \times \mathbf{blkstr}(2,k)$.

4: **ir – int32 scalar**

Default: The dimension of the array **b**.

r, the number of right-hand sides.

Constraint: $\mathbf{ir} > 0$.

5.3 Input Parameters Omitted from the MATLAB Interface

ldb

5.4 Output Parameters

1: **b(ldb,ir) – double array**

The array contains the n by r solution matrix X.

2: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, $\mathbf{n} < 1$,
or $\mathbf{nbloks} < 1$,
or $\mathbf{ir} < 1$,
or $\mathbf{ldb} < \mathbf{n}$,
or $\mathbf{n} < \mathbf{nbloks}$,
or **lena** is too small,
or illegal values detected in **blkstr**,
or **trans** $\neq$ 'N' or 'T'.

7 Accuracy

The accuracy of the computed solution depends on the conditioning of the original matrix A.

8 Further Comments

None.

9 Example

```
trans = 'N';
n = int32(18);
blkstr = [int32(2), int32(4), int32(5), int32(3), int32(4);
     int32(4), int32(7), int32(8), int32(6), int32(5);
     int32(3), int32(4), int32(2), int32(3), int32(0)];
a = [-1;
     -1;
     -0.98;
     0.25;
     -0.79;
     -0.87;
     -0.15;
     0.35;
     0.78;
     -0.82;
     -0.83;
     -0.21;
     0.31;
     0.12;
     -0.98;
     -0.93;
     -0.85;
     -0.01;
     -0.58;
     -0.84;
     0.89;
     0.75;
     0.04;
     0.37;
     -0.6899999999999999;
     0.32;
     0.87;
     -0.9399999999999999;
     -0.98;
     -1;
     0.38;
     -0.96;
     -0.76;
     -0.53;
     -1;
     -1;
     -0.99;
     -0.87;
     -0.93;
     0.85;
     0.17;
     -0.91;
     -0.14;
     -0.91;
     -0.39;
     -1.37;
     -0.28;
     -1;
     0.1;
     0.79;
     1.29;
```

```
     0.9;
     -0.59;
     -0.89;
     -0.71;
     -1.59;
     0.78;
     -0.99;
     -0.68;
     0.39;
     1.1;
     -0.93;
     0.21;
     -0.09;
     -0.99;
     -1.63;
     -0.76;
     -0.73;
     -0.58;
     -0.12;
     -1.01;
     0.48;
     -0.48;
     -0.21;
     -0.75;
     -0.27;
     0.08;
     -0.67;
     -0.24;
     0.61;
     0.5600000000000001;
     -0.41;
     0.54;
     -0.99;
     0.4;
     -0.41;
     0.16;
     -0.93;
     0.16;
     -0.16;
     0.7;
     -0.46;
     0.98;
     0.43;
     0.71;
     -0.47;
     -0.25;
     0.89;
     -0.97;
     -0.98;
     -0.92;
     -0.9399999999999999;
     -0.6;
     -0.73;
     -0.52;
     -0.54;
     -0.3;
     0.07000000000000001;
     -0.46;
     -1;
     0.18;
     0.04;
     -0.58;
     -0.36];
tol = 0;
b = [-2.92;
     -1.27;
     -1.3;
     -1.17;
     -2.1;
     -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];
[a, pivot, tol, index, ifail] = f01lh(n, blkstr, a, tol);
[bOut, ifail] = f04lh(trans, blkstr, a, pivot, b)
```

```
bOut =
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
ifail =
           0
```